StreamRunner™ AVA/ATV User's Manual

MANU0188-02 - Rev. A, 12/26/97

Software Version 5.0.x

FORE Systems, Inc.

1000 FORE Drive
Warrendale, PA 15086-7502
Phone: 412-742-4444
FAX: 412-742-7742

http://www.fore.com

Legal Notices

FCC CLASS A NOTICE

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void this user's authority to operate this equipment.

NOTE: The AVA-300 and ATV-300 have been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of the equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

DOC CLASS A NOTICE

This digital apparatus does not exceed Class A limits for radio noise emission for a digital device as set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class A prescrites dans le reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

CE *NOTICE*

Marking by the symbol **CE** indicates compliance of this system to the EMC (Electromagnetic Compatibility) directive of the European Community and compliance to the Low Voltage (Safety) Directive. Such marking is indicative that this system meets or exceeds the following technical standards:

- EN 55022 - "Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment."
- EN 50082-1 - "Electromagnetic compatibility - Generic immunity standard Part 1: Residential, commercial, and light industry."
- IEC 1000-4-2 - "Electromagnetic compatibility for industrial-process measurement and control equipment Part 2: Electrostatic discharge requirements."
- IEC 1000-4-3 - "Electromagnetic compatibility for industrial-process measurement and control equipment Part 3: Radiate electromagnetic field requirements."
- IEC 1000-4-4 - "Electromagnetic compatibility for industrial-process measurement and control equipment Part 4: Electrical fast transient/burst requirements."

SAFETY CERTIFICATIONS

Power supply is certified to meet Information Technology Equipment safety standards UL 1950, CSA 22.2 No. 950, and EN 60950.

Table of Contents

Preface

Video and Audio Over ATM

The intent of this manual is to supply users of FORE Systems' *StreamRunner*™ AVA / ATV hardware and software with all the necessary information to configure and use this equipment successfully. This document also provides general product information, network configuration information, and video/audio-over-ATM theory. This document was created for users with various levels of experience, but assumes a basic familiarity with ATM and video/audio concepts. Please read this document carefully before attempting to perform the procedures contained in it. If you have any questions or problems with installation or configuration after reading this manual, please contact FORE Systems' Technical Support.

Chapter Summaries

Chapter 1 - Introduction - Provides an overview of the ATM Standard and describes FORE Systems' *StreamRunner* AVA / ATV hardware and software.

Chapter 2 - The AVA-300 - Provides an overview of the *StreamRunner* AVA-300.

Chapter 3 - The ATV-300 - Provides an overview of the *StreamRunner* ATV-300.

Chapter 4 - UNIX Basic Setup - Describes how to get started with *StreamRunner* AVA / ATV hardware and software on UNIX-based platforms over an ATM network.

Chapter 5 - Windows NT/95 Basic Setup - Describes how to get started with *StreamRunner* AVA / ATV hardware and software on Windows NT- and / or 95-based platforms over an ATM network.

Chapter 6 - SVA Control - Provides an overview of SVA Control, the Windows NT graphical user interface for configuring managers, traders, and patches.

Chapter 7 - SVA Software - Provides an overview of the *StreamRunner* SVA Software and describes how to use it.

Chapter 8 - svc-rtds - Provides an overview of the *StreamRunner* SVA Software distribution's Real-Time Display Software (RTDS) and describes how to operate its graphical user interface.

Chapter 9 - Advanced Topics - Provides an overview of features that provide increased flexibility and functionality to the video / audio network.

Chapter 10 - Troubleshooting - Describes how to recognize, isolate, and resolve various problems associated with video and audio over ATM.

Appendix A - Manual and Reference Pages - Provides a reprint of the SVA on-line manual pages.

Appendix B - Environment Variables - Provides extended environment variable setup instructions.

Appendix C - PVC Control Channels - Describes how to configure PVC-based control channels.

Appendix D - ATV-300 Firmware Upgrade - Describes how to upgrade your ATV-300's firmware version.

Appendix E - svamgr: PVC Control Channels - Provides an explanation of PVC-based manager registration.

Appendix F - svamgr: Early AVA-300 Units - Describes how to configure signalling channels with devices carrying early firmware versions.

Appendix G - Audio Conference Setup - Describes how to configure an audio set-up for a high-quality bidirectional teleconference.

Appendix H - Windows NT/95: Manual De-install - Describes how to manually de-install the SVA Software distribution from your Windows NT/95 workstation.

Technical Support

In the U.S.A., you can contact FORE Systems' Technical Support using any one of the following methods:

1. If you have access to the Internet, you may contact FORE Systems' Technical Support via e-mail at:

   support@fore.com

2. You may telephone your questions to "support" at:

   800-671-FORE or 412-635-3700

3. You may FAX your questions to "support" at:

   412-742-7900

4. You may send questions, via U.S. Mail, to:

   FORE Systems, Inc.
   1000 FORE Drive
   Warrendale, PA 15086-7502

Technical support for non-U.S.A. customers should be handled through your local distributor.

No matter which method is used for support, please be prepared to provide your support contract ID number, the serial number(s) of the product(s), and as much information as possible describing your problem / question.

Typographical Styles

Throughout this manual, all specific commands meant to be entered by the user appear on a separate line in bold typeface. In addition, use of the `Enter` or `Return` key is represented as `<ENTER>`. The following example demonstrates this convention:

`cd /usr <ENTER>`

File names that appear within the text of this manual are represented in the following style: "...the `fore_install` program installs this distribution."

Command names that appear within the text of this manual are represented in the following style: "...using the **`flush-cache`** command clears the bridge cache."

Subsystem names that appear within the text of this manual are represented in the following style: "...to access the **`bridge`** subsystem..."

Parameter names that appear within the text of this manual are represented in the following style: "...using *`<seg-list>`* allows you to specify the segments for which you want to display the specified bridge statistics."

Any messages that appear on the screen during software installation and network interface administration are shown in `Courier` font to distinguish them from the rest of the text as follows:

```
.... Are all four conditions true?
```

Important Information Indicators

To call your attention to safety and otherwise important information that must be reviewed to ensure correct and complete installation, as well as to avoid damage to the FORE Systems product or to your system, FORE Systems utilizes the following *WARNING/CAUTION/NOTE* indicators.

WARNING statements contain information that is critical to the safety of the operator and/or the system. Do not proceed beyond a ***WARNING*** statement until the indicated conditions are fully understood or met. This information could prevent serious injury to the operator, damage to the FORE Systems product, the system, or currently loaded software, and is indicated as follows:

WARNING!

Hazardous voltages are present. To reduce the risk of electrical shock and danger to personal health, follow the instructions carefully.

CAUTION statements contain information that is important for proper installation/operation. Compliance with **CAUTION** statements can prevent possible equipment damage and/or loss of data and are indicated as follows:

CAUTION

You risk damaging your equipment and/or software if you do not follow these instructions.

NOTE statements contain information that has been found important enough to be called to the special attention of the operator and is set off from the text as follows:

NOTE

These `mute` and `pause` controls are local to the ATV-300. Their use does not effect an AVA-300 sending video or audio streams to the ATV-300.

Safety Precautions

For your protection, observe the following safety precautions when setting up equipment:

- Follow all warnings and instructions marked on the equipment.
- Ensure that the voltage and frequency of your power source matches the voltage and frequency inscribed on the equipment's electrical rating label.
- Never push objects of any kind through openings in the equipment. Dangerous voltages may be present. Conductive foreign objects could produce a short circuit that could cause fire, electric shock, or damage to your equipment.

Modifications to Equipment

Do not make mechanical or electrical modifications to the equipment. FORE Systems, Inc., is not responsible for regulatory compliance of a modified FORE product.

Placement of a FORE Systems Product

CAUTION

To ensure reliable operation of your FORE Systems product and to protect it from overheating, openings in the equipment must not be blocked or covered. A FORE Systems product should never be placed near a radiator or heat register.

Power Cord Connection

WARNING!

FORE Systems products are designed to work with single-phase power systems having a grounded neutral conductor. To reduce the risk of electrical shock, do not plug FORE Systems products into any other type of power system. Contact your facilities manager or a qualified electrician if you are not sure what type of power is supplied to your building.

WARNING!

Your FORE Systems product is shipped with a grounding type (3-wire) power cord. To reduce the risk of electric shock, always plug the cord into a grounded power outlet.

CHAPTER 1 Introduction

This chapter provides an introduction to the ATM Standard and FORE Systems' *StreamRunner* AVA / ATV hardware and software.

1.1 Overview of the ATM Standard

Asynchronous Transfer Mode, or ATM, is a communication architecture based on the switching of small, fixed-length packets of data called cells. In ATM, all data is transferred in 53-byte cells. Each cell has a 5-byte header that identifies the cell's route through the network and 48-bytes containing user data. The user data carries any headers or trailers required by higher-level protocols.

Perhaps the most important advantage offered by ATM, in addition to the speed at which data is transferred, is its open-ended growth path. ATM is not locked into a single, physical medium or speed. The fixed-size ATM cell allows traffic from multiple sources (simultaneous video, audio, and data communication) to be switched to multiple destinations by fast ATM switches.

For any two parties to communicate over an ATM network, a connection, either a Permanent Virtual Circuit (PVC) or a Switched Virtual Circuit (SVC), must be established between them. How such connections are set up depends on the particular ATM network in use. However, in all cases, an end-to-end ATM connection can be thought of as a set of sub-connections through the network.

Each sub-connection is defined by a Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) pair. Thus, an ATM connection is a set of VPI / VCI pairs, with one pair for each directly connected piece of equipment.

1.2 *StreamRunner* SVA Software

The *StreamRunner* SVA Software (hereafter referred to as the SVA Software) controls and manages the *StreamRunner* AVA-300 and the *StreamRunner* ATV-300, creating a distributed multimedia environment. The main components of the SVA Software distribution are:

svamgr Manages the AVA-300 and ATV-300, and implements UNI 3.0 or UNI 3.1 signalling on behalf of these devices.

svc-rtds Provides a Graphical User Interface (GUI) that allows you to query the ATM network for accessible video and audio sources to play, edit, and save on your UNIX or Windows NT/95 workstation.

svapatch Provides direct connection of video and audio streams between AVA-300s and ATV-300s.

In addition, the SVA Software consists of other applications, programs, and scripts that allow you to configure the AVA-300 and ATV-300 to your specific requirements. These features are discussed throughout this User's Manual and in Chapter 7.

1.3 *StreamRunner* AVA-300

The *StreamRunner* AVA-300 (hereafter referred to as the AVA-300) receives audio and video signals from conventional sources such as a VCR, camcorder, or stereo receiver, and digitizes these signals for transfer over an ATM network. The AVA-300 supports PAL and NTSC video formats for optimal workstation reception and Motion-JPEG compression for real-time video applications.

1.4 *StreamRunner* ATV-300

The *StreamRunner* ATV-300 (hereafter referred to as the ATV-300) decodes digital signals transmitted over the ATM network by an AVA-300 and converts them into conventional analog signals. The ATV-300 provides full-frame-rate interlaced video output with up to DAT-quality audio. The ATV-300 can decode multiple video streams from the ATM network and display them in a tiled or picture-in-picture presentation.

The ATV-300 also supports an infrared remote control facility which is capable of interacting with software running remotely in the network.

CHAPTER 2 The AVA-300

This chapter provides an overview of the AVA-300. Information about unpacking, general features, technical and physical specifications, and front and rear panel details is included.

2.1 Unpacking Information

Before installing the AVA-300, inspect the package for any damage that may have occurred during shipping. If the package shows any sign of external damage or rough handling, notify your carrier's representative. When unpacking the AVA-300, be sure to keep all original packing materials for storing, transporting, or returning the product.

CAUTION

All products returned to FORE Systems under warranty must be packed in their original packing materials.

Verify the contents of the package. The following items should be present:

- AVA-300
- External Power Supply Unit
- Power Cable, IEC / US / Continental
- Release Notes (if applicable)
- *StreamRunner* AVA / ATV User's Manual (this manual)

If any of the above items is missing or damaged, please contact FORE Systems' Technical Support or your local distributor immediately. Before proceeding with any installation, please read the enclosed release notes that may accompany the unit.

2.2 AVA-300 Overview

As new applications drive the requirement for more bandwidth, users are turning to ATM to provide a method for implementing high quality video transfer. The AVA-300 provides an efficient platform for one-way video and audio multicasts. The AVA-300 is a stand-alone device that is suitable for a wide variety of applications.

The output from a camcorder, VCR, or any other standard video and/or audio source may be directly connected to an ATM network through the AVA-300. The AVA-300 converts video and audio inputs from their analog format to an uncompressed or compressed digital format encoded over an ATM cell stream. The ATM network can then be used to switch or multicast the video and audio to any number of desired locations.

The only additional hardware required to display video and playback audio on a UNIX workstation or Windows NT/95 workstation is an ATM network interface card. In the very near future, it is speculated that many vendors will supply workstations and PCs with an ATM network interface as standard. Also, the capability of workstations is improving to the extent that the video performance currently attainable by software decode could only recently be achieved by specialized third-party video boards.

2.3 AVA-300 Technical and Physical Specifications

The Technical Specifications of the AVA-300 are detailed in Table 2.1:

Table 2.1 - AVA-300 Technical Specifications

Feature	Specification
ATM Interface	155 Mbps UTP (RJ-45 connectors) 155 Mbps MMF (SC connectors) 155 Mbps intermediate reach SMF (SC connectors)
Video Formats	PAL (50Hz) and NTSC (60Hz), software selectable
Video Connectors	6 RCA/Phono sockets configurable as 6 composite channels or 3 S-Video channels; video inputs may be multiplexed onto a maximum of 4 video output streams
Digital Video	Uncompressed 24-bit, 16-bit, or 8-bit RGB or 8-bit mono; or compressed using Motion JPEG with software-selectable compression factors
Video Sizing	Analog sampling region and digital display size configurable on a per-stream basis (down-scaling of sampled region to fit display size)
Video Controls	Input brightness, contrast, and color all controllable over network
Video Performance	50 fields per second (PAL) or 60 fields per second (NTSC)
Audio Connectors	6 RCA/Phono input sockets equalling 3 stereo input channels; one input channel selectable for ATM network transmission at a time
Digital Audio	8- or 16-bit PCM, A-Law or μ-Law (stereo/mono) formats; sampling rate from 5kHz to 44.1kHz (CD) and 48kHz (DAT)
Rate Control	Software configurable cell pacing for ATM output streams to ensure traffic contract compliance
Serial Interface	RS-232, accessible over the ATM network
Auxiliary Connector	External Configuration Module (ECM) plug-in stores video and audio stream configurations for power-up auto-loading
ATM Protocols	ATM Forum's UNI 3.0 and UNI 3.1 supported by accompanying SVA 5.0 software
Emissions	FCC Part 15, Class A; CISPR 22, Class A; VCCI Class 1
Safety	US: UL1950; Canada: CSA 22.2; No. 950-M89; Europe: EN60950

The Physical Specifications for AVA-300s with a hardware version less than 300.10 are detailed in Table 2.2:

Table 2.2 - Physical Specifications for AVA-300 Hardware Version less than 300.10

Feature	Specification
Dimensions	H: 2.4” (6 cm) x W: 10.6” (27 cm) x D: 9.8” (25 cm)
Weight	3.8 lb. (1.8 kg)
Operating Temperature	32° F to 104° F (0° C to 40°C)
Operating Humidity	10% to 90% non-condensing
Power	External PSU: 120/240Vac, 50/60 Hz (autoranging) Tolerance: 88-132Vac/176-264Vac, 47-63 Hz Maximum Power Consumption: 25 W

The Physical Specifications for AVA-300s with a hardware version of 300.10 or later are detailed in Table 2.3:

Table 2.3 - Physical Specifications for AVA-300 Hardware Version 300.10 or later

Feature	Specification
Dimensions	H: 2.4” (6 cm) x W: 10.6” (27 cm) x D: 9.8” (25 cm)
Weight	3.8 lb. (1.8 kg)
Operating Temperature	32° F to 104° F (0° C to 40°C)
Operating Humidity	10% to 90% non-condensing
Power	External PSU: 120/240Vac, 50/60 Hz (autoranging) Tolerance: 88-132Vac/176-264Vac, 47-63 Hz Maximum Power Consumption: 25 W

2.4 AVA-300 Front Panel Detail

The AVA-300's physical front panel features, including connection sockets, indicator lights, and their functions, are described in this section. Figure 2.1 shows the AVA-300's front panel arrangement, and is followed by an explanation of each feature.

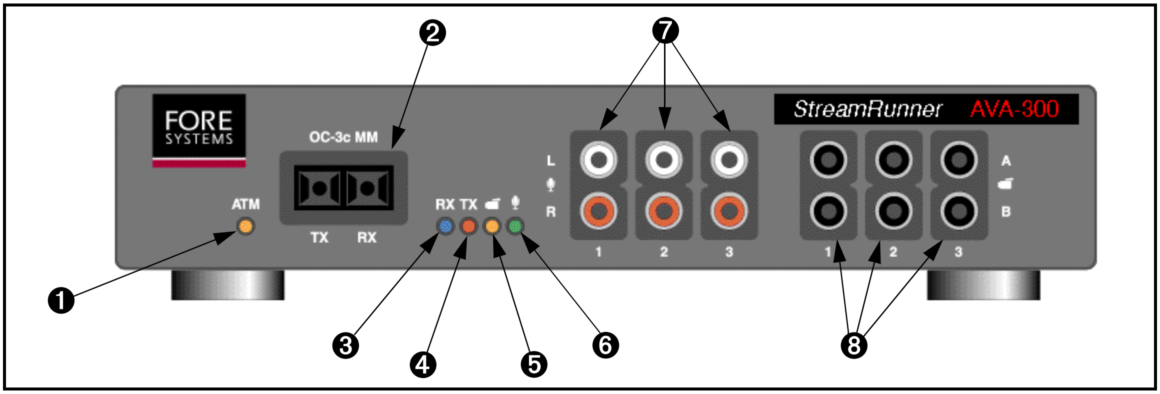

Figure 2.1 - AVA-300 Front Panel Detail

ATM LED ❶ Indicates the status of ATM network connectivity. **Green** indicates connectivity to the network. **Red** indicates connectivity has not been achieved.

ATM Interface ❷ Provides connectivity to the ATM network. Figure 2.1 illustrates an AVA-300 with a fiber optic interface.

RX LED ❸ Indicates that cells are being received from your switch by the AVA-300 when it blinks **blue**.

TX LED ❹ Indicates that cells are being transmitted by the AVA-300 to the switch when it blinks **red**.

Video LED ❺ Indicates that video signals are being digitized by the AVA-300 when it is solid or blinks **yellow**.

Audio LED ❻ Indicates that audio signals are being digitized by the AVA-300 when it is solid **green**.

Audio Connectors ❼ Provide three stereo input channels, one of which may be selected for ATM network transmission at a time.

Video Connectors ❽ Provide three S-Video or six composite video input channels. Inputs may be multiplexed onto a maximum of four video output streams.

2.5 AVA-300 Rear Panel Detail

The AVA-300's physical rear panel features are described in this section. Figure 2.2 shows the AVA-300's rear panel arrangement, and is followed by an explanation of each feature.

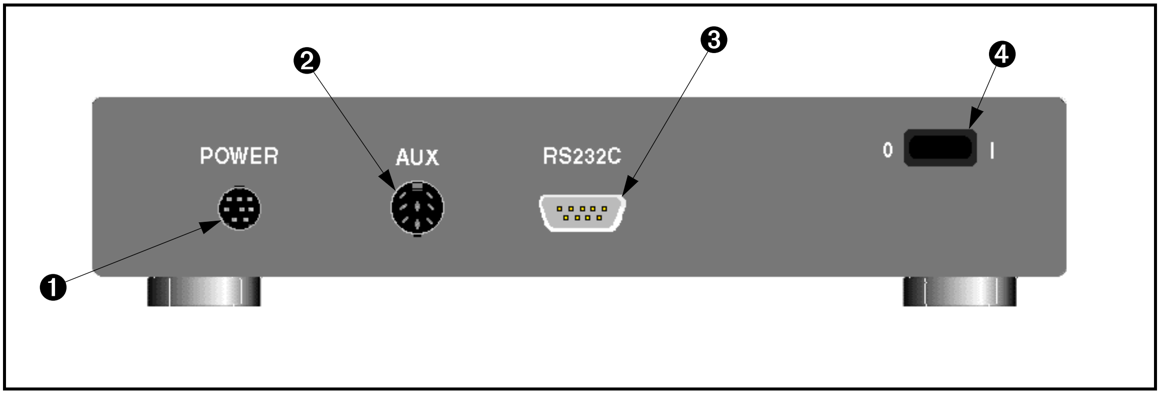

Figure 2.2 - AVA-300 Rear Panel Detail

Power DIN Connector ❶	Provides External Power Supply Unit (PSU) connectivity.
AUX DIN Connector ❷	Provides External Configuration Module (ECM) connectivity.
RS232C Port ❸	Provides serial connectivity to and from the ATM network.
Power Switch ❹	Powers the AVA-300 on or off (see note below).

NOTE

Some AVA-300 models (hardware version 300.10 and later) do not have a power switch. These units may be switched on or off at the main supply.

2.5.1 AVA-300 Serial Port Hardware

The AVA-300 RS232C serial port, located on the unit's rear panel, is a 9-pin male D-connector. The serial port's pinout is shown in Figure 2.3.

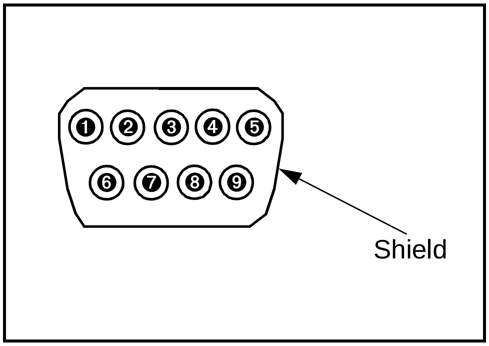

Figure 2.3 - AVA-300 Serial Port Pinout

The individual pins and cable shield have the following functions:

Shield	Ground
Pin ❷	Receive data to the AVA-300
Pin ❸	Transmit data from the AVA-300
Pin ❹	Tied high (to 9V)
Pin ❺	Ground
Pin ❼	Tied high (to 9V)

The remaining pins are unconnected.

NOTE A null modem (cross-over) cable is required to connect to a workstation or similar device. Hardware flow-control (RTS/CTS-type schemes) are not supported.

2.6 External Power Supply Unit (PSU)

The AVA-300 and ATV-300 external Power Supply Unit (PSU) is shown in Figure 2.4. The PSU is a step-down transformer that automatically senses the correct line voltage. The PSU attaches to the Power DIN connector located on the AVA-300's rear panel. The power cable attaches to the PSU's IEC connector port.

Some AVA-300 and ATV-300 models are shipped with a slightly different power supply.

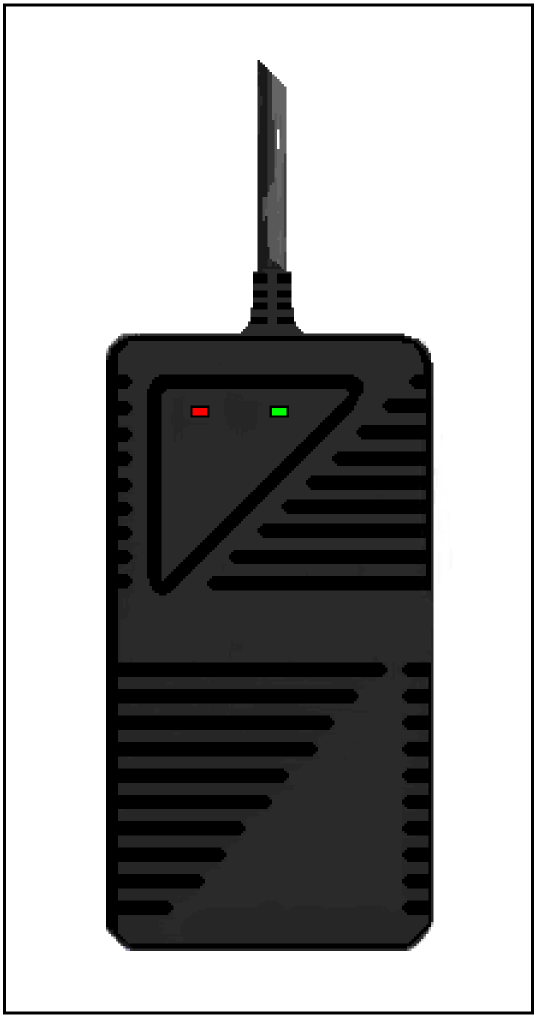

Figure 2.4 - External Power Supply Unit (PSU)

CHAPTER 3 The ATV-300

This chapter provides an overview of the ATV-300. Information about unpacking, general features, technical and physical specifications, front and rear panel details, and the remote-controlled menu system is included.

3.1 Unpacking Information

Before installing the ATV-300, inspect the package for any damage that may have occurred during shipping. If the package shows any sign of external damage or rough handling, notify your carrier's representative. When unpacking the ATV-300, be sure to keep all original packing materials for storing, transporting, or returning the product.

CAUTION

All products returned to FORE Systems under warranty must be packed in their original packing materials.

Verify the contents of the package. The following items should be present:

- ATV-300
- ATV-300 Infrared Remote Control Unit
- External Power Supply Unit
- Power Cable, IEC / US / Continental
- Release Notes (if applicable)
- *StreamRunner* AVA / ATV User's Manual (this manual)

If any of the above items is missing or damaged, please contact FORE Systems' Technical Support or your local distributor immediately. Before proceeding with any installation, please read the enclosed release notes that may accompany the unit.

3.2 ATV-300 Overview

ATM provides both greater bandwidth and a method for implementing high-quality video and audio transfer. The ATV-300, when used together with the AVA-300, provides a flexible, high-performance solution for video and audio signal transmission over an ATM network.

The AVA-300 digitizes video and audio signals for direct transmission onto an ATM network. These digital media streams may be received and processed by UNIX workstations and Windows NT-equipped PCs that are directly connected to the ATM network. They may also be received and processed by the ATV-300.

The ATV-300 is a dedicated unit for receiving and decoding the digital streams generated by an AVA-300, either via an ATM network or by direct connection. The ATV-300 is suitable for situations in which high-quality output signals are required or in which it would be unsuitable to place a desktop computer.

3.3 ATV-300 Technical and Physical Specifications

The Technical Specifications of the ATV-300 are detailed in Table 3.1:

Table 3.1 - ATV-300 Technical Specifications

Feature	Specification
ATM Interface	155 Mbps UTP (RJ-45 connectors) 155 Mbps MMF (SC connectors) 155 Mbps intermediate reach SMF (SC connectors)
Video Formats	PAL (50 Hz) and NTSC (60 Hz), software selectable
Digital Video	Concurrent decompression of up to 4 multiple AVA format Motion JPEG digital video streams Picture-in-picture and tiled video presentation
Audio Connectors	2 RCA / Phono sockets for stereo output
Digital Audio	8- or 16-bit PCM, A-Law or μ-Law (stereo / mono) formats; sampling rate from 5kHz to 44.1kHz (CD) and 48kHz (DAT)
User Control	Infrared remote controlled on-screen menu system
Auxiliary Connector	External Configuration Module (ECM) plug-in stores video and audio stream configurations for power-up auto-loading
ATM Protocols	ATM Forum's UNI 3.0 and UNI 3.1 supported by accompanying SVA software
Emissions	FCC Part 15, Class A; CISPR 22, Class A; VCCI Class 1
Safety	US: UL1950; Canada: CSA 22.2; No. 950-M89; Europe: EN60950

The Physical Specifications for ATV-300s with a hardware version less than 300.10 are detailed in Table 3.2:

Table 3.2 - Physical Specifications for ATV-300 Hardware Version less than 300.10

Feature	Specification
Video Connectors	2 RCA / Phono sockets for S-Video output 2 RCA / Phono sockets for duplicated composite signal
Dimensions	H: 2.4” (6 cm) x W: 10.6” (27 cm) x D: 9.8” (25 cm)
Weight	3.8 lb. (1.8 kg)
Operating Temperature	32° F to 104° F (0° C to 40°C)
Operating Humidity	10% to 90% non-condensing
Power	External PSU: 120 / 240Vac, 50 / 60 Hz (autoranging) Tolerance: 88-132Vac / 176-264Vac, 47-63 Hz Maximum Power Consumption: 25 W

The Physical Specifications for ATV-300s with a hardware version of 300.10 or later are detailed in Table 3.3:

Table 3.3 - Physical Specifications for ATV-300 Hardware Version 300.10 or later

Feature	Specification
Video Connectors	1 S-Video connector for S-Video output 1 RCA / Phono socket for composite output
Dimensions	H: 2.36” (6 cm) x W: 9.945” (25.24 cm) x D: 9.35” (23.77 cm)
Weight	3.8 lb. (1.8 kg)
Operating Temperature	32° F to 104° F (0° C to 40°C)
Operating Humidity	10% to 90% non-condensing
Power	External PSU: 115 / 230Vac (normal voltage), 100Vac to 250Vac (voltage range) 47 Hz to 63 Hz (frequency) Maximum Power Consumption: 25 W

3.4 ATV-300 Front Panel Detail

The ATV-300's physical front panel features, including connection sockets, indicator lights, and their functions, are described in this section. Figure 3.1 shows the ATV-300's front panel arrangement, and is followed by an explanation of each feature.

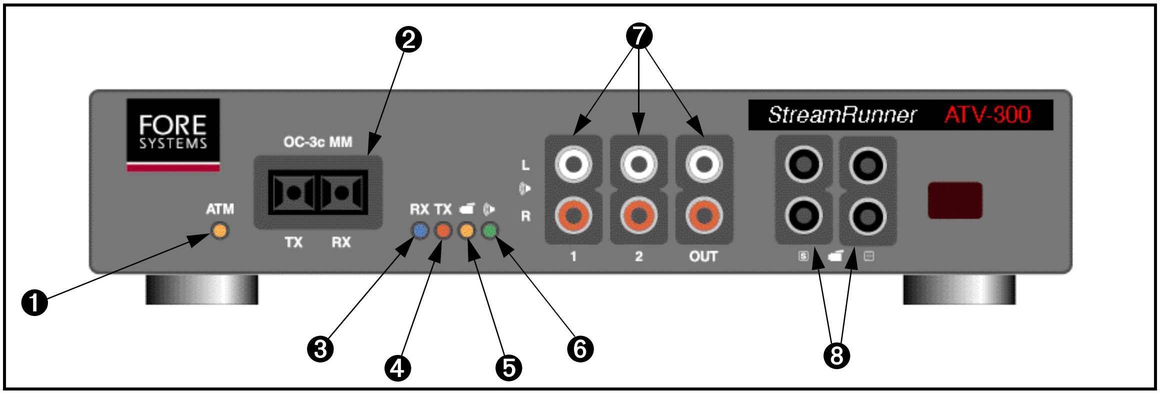

Figure 3.1 - ATV-300 Front Panel Detail

ATM LED ❶ Indicates the status of ATM network connectivity. **Green** indicates connectivity to the network. **Red** indicates connectivity has not been achieved.

ATM Interface ❷ Provides connectivity to the ATM network. Figure 3.1 illustrates an ATV-300 with a fiber optic interface.

RX LED ❸ Indicates that cells are being received from your switch by the ATV-300 when it blinks **blue**.

TX LED ❹ Indicates that cells are being transmitted through the ATV-300 when it blinks **red**.

Video LED ❺ Indicates that video signals are being sent through the ATV-300 when it is solid or blinks **yellow**.

Audio LED ❻ Indicates that audio signals are being sent through the ATV-300 when it is solid **green**.

Audio Connectors ❼ Provide two stereo input channels and one stereo output channel.

Video Connectors ❽ Provide one S-Video output channel (`Top` chrominance output, `Bottom` luminance output) and two composite video output channels.

The ATV-300

Some versions of the ATV-300 (hardware version 300.10 and later) have a slightly different front panel arrangement, as illustrated in Figure 3.2.

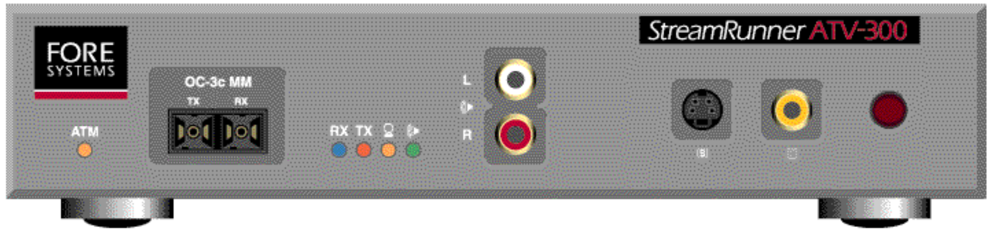

Figure 3.2 - ATV-300 Front Panel Detail, Hardware Version 300.10 and Later

These ATV-300s are functionally equivalent to the earlier units, but include the following front panel modifications:

- The unsupported stereo audio input connectors have been removed.
- A single composite video RCA / Phono output rather than two duplicated connectors.
- A standard S-Video connector rather than two separate RCA / phono sockets.

3.5 ATV-300 Rear Panel Detail

The ATV-300's physical rear panel features are described in this section. Figure 3.3 shows the ATV-300's rear panel arrangement, and is followed by an explanation of each feature.

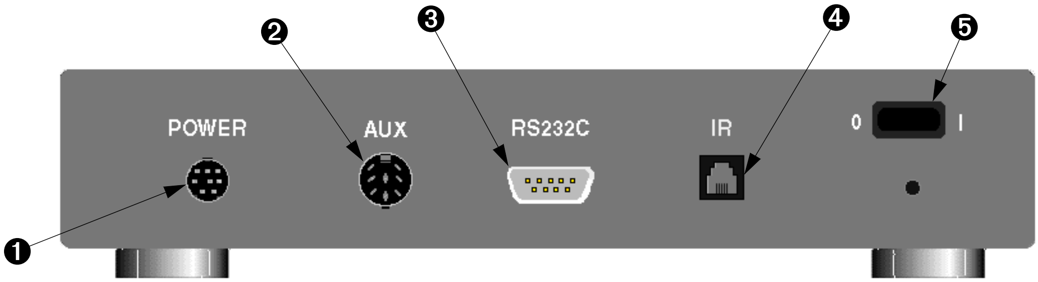

Figure 3.3 - ATV-300 Rear Panel Detail

Power DIN Connector ❶ Provides External Power Supply Unit (PSU) connectivity.

AUX DIN Connector ❷ Provides External Configuration Module (ECM) connectivity.

RS232C Port ❸ Provides engineering testing connectivity (not a user-supported feature).

IR Modular Pin Jack ❹ Not a user-supported feature (see note below).

Power Switch ❺ Powers the ATV-300 on or off (see note below).

NOTE

Some ATV-300 models (hardware version 300.10 and later) have neither an IR modular pin jack nor a power switch. These units may be switched on or off at the main supply.

3.6 External Power Supply Unit (PSU)

The ATV-300 uses the same external Power Supply Unit (PSU) that is used with the AVA-300. Please refer to Section 2.6 for information on and illustrations of the applicable PSUs.

3.7 ATV-300 Infrared Remote Control Unit

The ATV-300 Infrared Remote Control Unit, shown in Figure 3.4, gives you access to the Graphical User Interface (GUI) which may be used to control the operation of the ATV-300. The Remote uses line-of-sight transmission, much like the infrared controls supplied with TVs and VCRs. To maximize efficiency when using the Remote, you should aim for the infrared receiver eye located on the extreme right of the ATV-300's front panel.

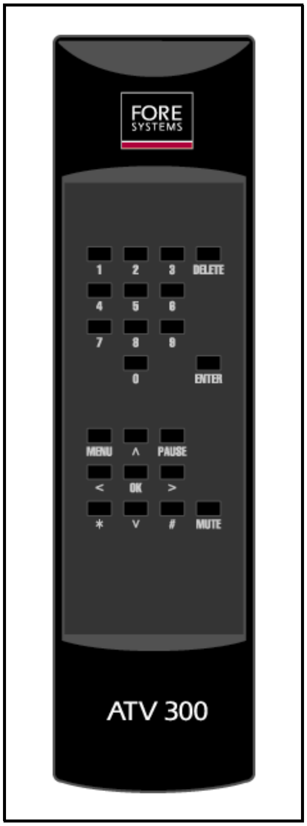

Figure 3.4 - ATV-300 Infrared Remote Control Unit

3.8 ATV-300 Interface Control

The ATV-300's interface consists of three types of components:

- Menus
- Numeric input boxes
- Slider bars

If you make an incorrect entry or operation, an error box will be displayed, providing you with information about how to correct your entry/operation. The methods by which menus, numeric input boxes, and slider bars are controlled with the Remote are described below.

3.8.1 Menus

The `menu` button on the Remote is used to display the ATV-300 `Main Menu` (described in Section 3.9) or to remove a submenu from the display. Some menus are **modal,** which means you are required to select an item from that menu rather than remove the menu using the `menu` button.

When you display a menu, a white ◆ indicator will be positioned to the left of one of the menu items. Pressing the ∧ or ∨ button on the Remote moves the ◆ up or down one item, and pressing `OK` or `<Enter>` selects the item it is alongside.

Selecting a menu item causes an associated operation to be performed, or causes another menu, numeric input box, or slider bar to be displayed. In the case in which another GUI component is displayed, pressing the > button does the same thing as `OK` or `<Enter>`, and pressing the < button returns you to the previous menu.

3.8.2 Numeric Input Boxes

Certain operations, such as designating a VCI, require you to input a numeric value. You do this by using a **numeric input box**. A brief description of the value to be entered is displayed, followed by the range of valid values in parentheses, followed by the current or default value. A pair of vertical arrows (↑ and ↓) is displayed on the right-hand side of the box.

Pressing the ∧ or ∨ button on the Remote increases or decreases the value being displayed in the box. When the desired value is reached, pressing `OK` or `<Enter>` confirms that value. The < button returns you to the previous menu or input box, and the > button moves you to the next menu or input box (if applicable).

You may also enter a value using the `digit` buttons `0` through `9` rather than using the ∧ or ∨ to get to the appropriate value. The `delete` button may be used to remove the rightmost digit of the current value.

NOTE

There may be situations in which the displayed value temporarily drops below the minimum-allowed value. In these cases, the background and outline of the numeric input box turns red and the value cannot be selected until the current value moves back into the valid range. It is impossible to enter a value which is greater than the maximum allowed.

To leave the current operation while in a numeric input box, press the `menu` button to remove the box and display the ATV-300 `Main Menu`.

NOTE

Some numeric input boxes require a value to be entered and cannot be removed by pressing the `menu` button. (These are modal fields.)

3.8.3 Slider Bars

A **slider bar** works in much the same way as a numeric input box, but is used in situations in which it is useful to see the current value as it relates to the maximum setting (such as changing the volume level). A red bar on a gray background is used to represent this relationship. In most cases, the effect of altering a slider bar setting is immediately visible or audible while the adjustment is being made.

Pressing the ∧ or ∨ button increases or decreases the current value. To confirm the current slider setting, press `OK` or `<Enter>`.

Pressing the `menu` button removes the slider bar and displays the ATV-300 `Main Menu`, pressing < returns you to the previous menu, and pressing `delete` restores the slider setting to its default value. Pressing the `digit` buttons `0` through `9` has no effect on a slider bar. Figure 3.5 illustrates a typical slider bar.

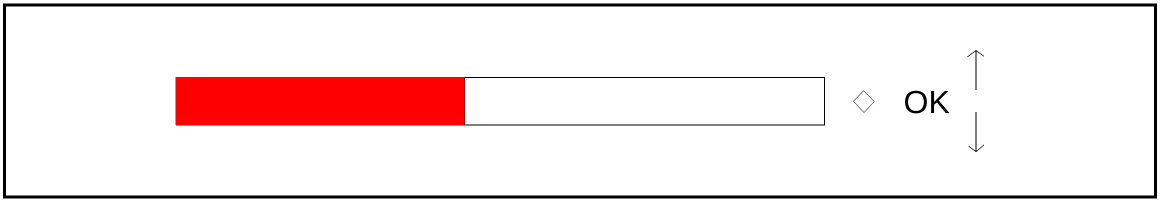

Figure 3.5 - Slider Bar

3.8.4 Mute and Pause Controls

Two Remote buttons, `mute` and `pause`, allow you to globally disable or re-enable all audio and/or video streams. Pressing `mute` once disables all audio output from the ATV-300. While the ATV-300 is in this state, the first icon shown in Figure 3.6 is displayed in the bottom-right corner of the display. Pressing `mute` again re-enables audio output. Similarly, pressing `pause` once stops all video streams, displaying the middle icon shown in Figure 3.6, and pressing it again restarts them. If audio and video are both disabled, the combined icon shown on the right in Figure 3.6 is displayed.

NOTE

These `mute` and `pause` controls are local to the ATV-300. Their use does not effect an AVA-300 sending video or audio streams to the ATV-300.

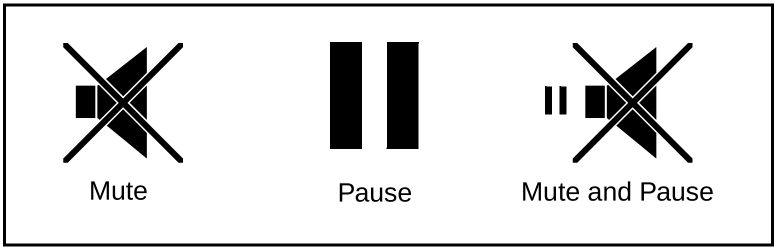

Figure 3.6 - Mute and Pause Icons

It is possible to disable the effect of these buttons to avoid streams being stopped inadvertently (see Section 3.9.3.7).

3.8.5 Power-Up Configuration Menu and Video Mode

When powered-up, the ATV-300 displays the Power-up `Configuration` menu, similar to the illustration shown in Figure 3.7.

ATV-300 configuration

Hardware version:	300.5
Firmware version:	300.18
Boot firmware:	1.7
Serial number:	9700000
Video capacity:	PAL, NTSC
Default video mode:	NTSC
Current video mode:	NTSC
Network interface:	OC3c 155
View stream information	

Figure 3.7 - ATV-300 Power-up Configuration Menu

NOTE

The Power-up `Configuration` menu will not be displayed if the ATV-300 has an External Configuration Module (ECM) attached which contains valid stream settings. At start-up, the stream configuration will immediately be restored.

The `Power-up Configuration` menu is removed if the ATV-300 is reset by remote SVA software (e.g., `svarun`.)

3.8.5.1 Selecting NTSC or PAL at Power-Up

Pressing the ✱ button while the `Power-up Configuration` menu is being displayed sets the ATV-300 to PAL output mode and removes the `Power-up Configuration` menu from the screen. PAL becomes the default mode on future power-ups or resets.

Similarly, pressing the # button while the `Power-up Configuration` menu is being displayed sets the ATV-300 to NTSC output mode and removes the `Power-up Configuration` menu from the screen. NTSC becomes the default mode on future power-ups or resets.

NOTE

If the ATV-300 is already in the required video output mode, pressing the corresponding button will have no effect. This option is provided so that if an ATV-300 is connected to a Video Display Unit (VDU) which is incapable of displaying the current default output (seen as a blank or scrolling image), the mode can be changed easily without any need to see what is being displayed.

3.9 The ATV-300 Main Menu

The ATV-300 `Main Menu`, shown in Figure 3.8, is the top-level of the ATV-300 system interface. The `Main Menu` provides the gateway to various options and submenus which allow you to examine and alter the ATV-300 configuration and to perform limited single-stream communication with a single AVA-300.

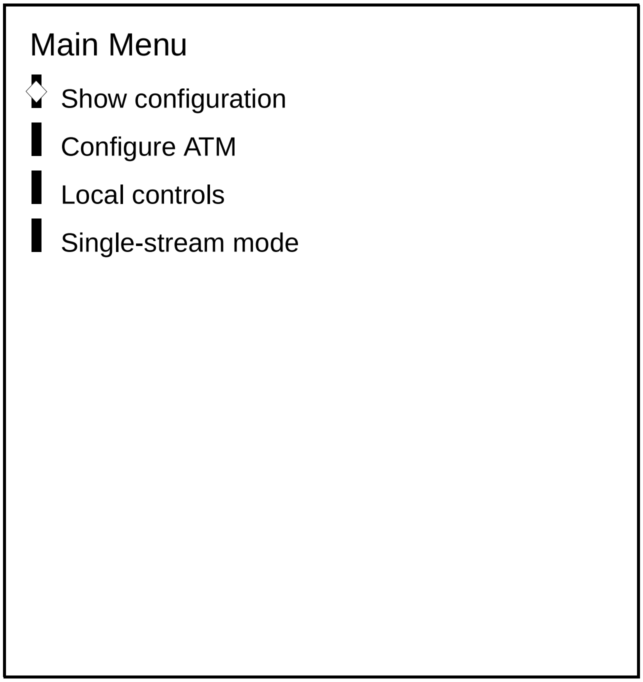

Figure 3.8 - The ATV-300 Main Menu

NOTE

Unless otherwise indicated, any configuration changes made only remain in effect until the ATV-300 is reset again. To make a permanent change, use the `Save User Preferences` option, as described in Section 3.9.3.4.

3.9.1 Show Configuration

Selecting the `Show Configuration` option displays a submenu consisting of two parts, as shown in Figure 3.9.

ATV-300 configuration

Hardware version:	300.5
Firmware version:	300.18
Boot firmware:	1.7
Serial number:	9700000
Video capacity:	PAL, NTSC
Default video mode:	NTSC
Current video mode:	NTSC
Network interface:	OC3c 155
View stream information	

Figure 3.9 - The Show Configuration Submenu

The top section displays the current settings for various ATV parameters, similar to the `Power-up Configuration` menu displayed when the ATV-300 is powered-up. The second item, `View Stream Information`, displays a list of the names of all currently-configured streams, or indicates that none are configured. Selecting one of those streams displays information on the characteristics of the stream, including:

- VCI number
- Pack factor
- Q-factor (video streams)
- Sample rate (audio streams)

3.9.2 Configure ATM

The `Configure ATM` option displays a submenu indicating the ATV-300's network interface type and the current setting of any configurable network interface parameters, as shown in Figure 3.10.

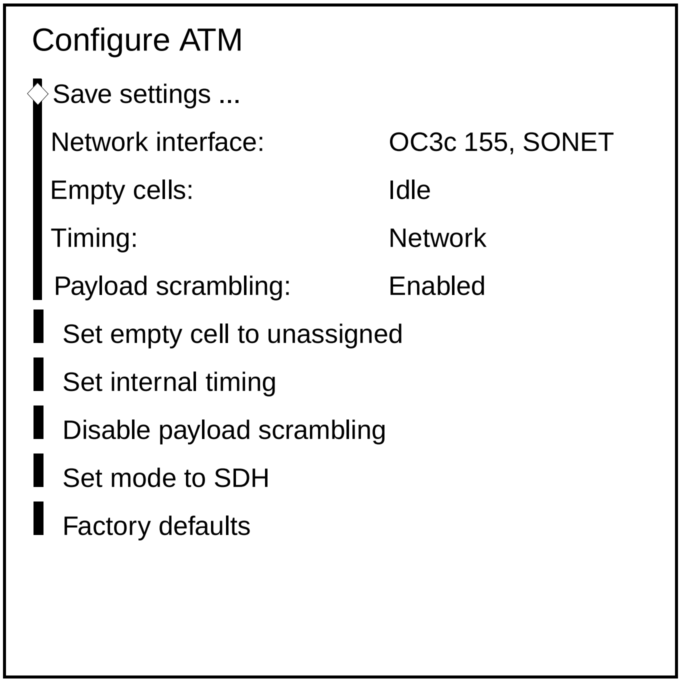

Figure 3.10 - The Configure ATM Submenu

If any parameters are configurable, further menu items will allow you to change those settings. For example, an ATM 155 Mbps unit would allow payload scrambling on the transmission interface to be enabled or disabled.

If you alter any configurable parameters, a confirmation menu is displayed after you remove the `Configure ATM` menu by pressing the `menu` or < button. The `Configure ATM` menu reminds you that changes made to the ATM configuration are not permanent, and gives you the option of saving the current user preferences to make them permanent.

NOTE

Selecting this option saves all user preferences, including display settings and the GUI configuration, not just the ATM configuration parameters.

3.9.3 Local Controls

Selecting `Local Controls` allows you to customize many ATV-300 characteristics, including the menu overlay, video mode, stream diagnostics, and the Remote, as shown in Figure 3.11.

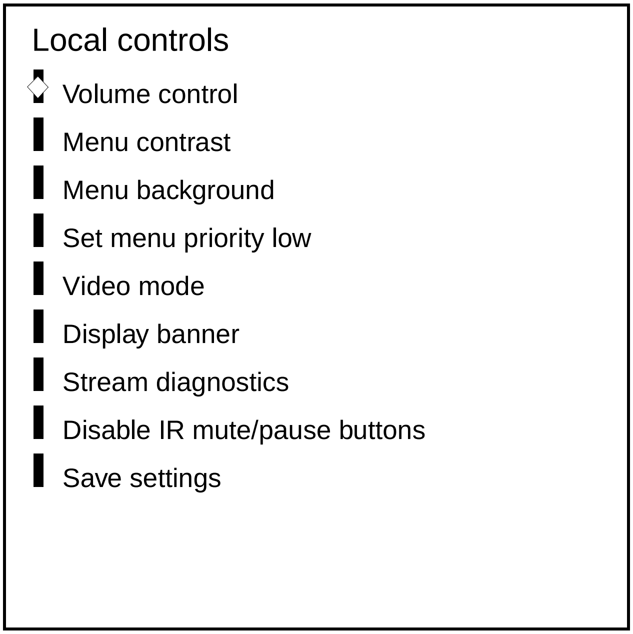

Figure 3.11 - The Local Controls Submenu

3.9.3.1 Volume Control

The `Volume Control` option displays a slider bar that you can use to alter the volume level of the audio output from the ATV-300.

3.9.3.2 Menu Contrast and Menu Background

The `Menu Contrast` option displays a slider bar that you can use to alter the contrast of the GUI components. `Menu Contrast` only affects GUI components and does not alter the appearance of any video streams.

The `Menu Background` option allows you to select the degree of transparency of the menu background color ranging from completely transparent to completely opaque.

3.9.3.3 Set Menu Priority

The priority taken by GUI operations over the processing of video and audio streams may be toggled between high and low:

Set Menu Priority High GUI operation receives a higher priority over video and audio stream processing. This ensures good GUI response, but may cause temporary video and audio stream pause when you are using the GUI.

Set Menu Priority Low Video and audio stream reproduction receives priority over GUI operation. This may result in slower GUI response when the ATV-300 is under heavy load from incoming ATM data. This is the default and recommended setting.

3.9.3.4 Video Mode

The ATV-300 provides video output in either `PAL` (50Hz, 625 lines) or `NTSC` (60Hz, 525 lines) format. The `Video Mode` option displays a menu that allows you to set the current output mode, the default output mode to be used on reset, and the video capability of the display device being used.

The first option on the `Video Mode` submenu, `Set Video Mode to PAL` (or `NTSC`), allows you to set the current and default video modes to `PAL` or `NTSC.` When you update the video format, the message screen shown in Figure 3.12 is displayed.

The video mode has been changed to PAL. This change is not permanent and will be lost when the unit is next reset. To make the change permanent, choose “Save user preferences” below.

Save user preferences

Continue without saving

Figure 3.12 - Set Video Mode Message Screen

As stated in this screen, you may either save the updated setting, or continue without saving, thereby reverting back to the original video mode the next time the ATV-300 is powered-up.

NOTE

In addition to the video mode settings, the `Set Video Mode to NTSC` (or `PAL`) option saves all other user preferences. Changing the current video mode causes the current ATV-300 stream configuration to be lost.

The second option on the `Video Mode` submenu, `Further Options`, allows the current and default modes to be altered individually and the video capability of the display device being used to be set. Setting the video capability has no direct effect on the operation of the ATV-300, but it does allow the ATV-300 to prompt for confirmation when a request is made to change to a video mode which the display device may not be able to support. If you attempt to change the current or default video mode to one which is not supported, the ATV-300 indicates this and suggests that you check the video output capability setting.

NOTE

Changing the default mode or video capability automatically saves all current user preferences without further prompting.

On power-up, if the default video mode of the unit is found to be one which is not supported by the display device, the ✳ and # buttons may be used to change the video mode.

3.9.3.5 Display (or Remove) Banner

Invoking the `Display Banner` option displays the following text near the bottom of the screen:

```
ATV-300 - Live ATM Video
```

The banner is useful during demonstrations. Selecting `Remove Banner` deletes it.

3.9.3.6 Stream Diagnostics

If a stream error occurs when the ATV-300 is displaying a video stream, an error box is overlaid on the stream indicating the type of error which has occurred; e.g., a size mismatch between the stream which the ATV-300 expects and the stream actually arriving. The `Stream Diagnostics` option allows you to specify whether such errors will be reported.

Selecting `Enable` (or `Disable`) `Stream Diagnostics` toggles the current state of error reporting. When error reporting is enabled, the length of time after an error has cleared until the error box is removed may be set using the `Set Delay Before Removal` option. Error boxes may be set for removal after 1, 2, or 5 seconds.

NOTE

In single-stream mode all error reporting is disabled, irrespective of the state which you have configured.

3.9.3.7 IR Mute—Pause Buttons

This option may be used to disable and re-enable the `mute` and `pause` buttons on the remote control when not in use. This is useful in order to avoid streams being stopped accidentally.

If mute or pause is in operation (indicated by an icon in the lower right corner of the display) when `Disable IR Mute—Pause Buttons` is selected, you are reminded of this and asked if you wish to switch off `mute` and `pause` before disabling the IR buttons. When `mute` and `pause` are not in operation no confirmation is requested.

3.9.3.8 Save Settings

If an External Configuration Module (ECM) is attached to the ATV-300, you may save the current stream settings to the ECM. An ATV-300 which is then powered-up or reset while that ECM is attached will automatically reconfigure itself with the same stream settings. The first item on the `Save Settings` menu performs this stream saving to the ECM.

The second item, `Save User Preferences`, internally saves the values of all user-configurable options, such as the current audio volume, display mode, and any network interface options (this does not require an ECM to be attached). If this is not used after changing the unit's configuration, the changes will generally be lost at the next reset or power-down (the few exceptions to this rule are mentioned specifically in the above text).

3.9.4 Single-stream mode

The Single-stream mode option lets you create a single video and a single audio stream between an ATV-300 and an AVA-300. This option is discussed in detail in Chapter 9 in this User's Manual.

CHAPTER 4 UNIX Basic Setup

This chapter is a step-by-step guide to get you started with your *StreamRunner* AVA/ATV hardware and SVA software system over an ATM network using a UNIX platform. Included are an overview of the system components and how to configure them, and how to install, configure, and use the software. This chapter assumes:

- A working knowledge of UNIX
- A basic knowledge of ATM networking concepts
- ATM connectivity using UNI 3.0 or UNI 3.1 signalling

The Basic Setup involves the following steps, each of which is detailed in this chapter:

- Installing the SVA Software
- Configuring the Hardware
- Verifying Proper Setup
- Setting up a Trader and Manager
- Running the Real-Time Display Software (`svc-rtds`)
- Running `svapatch` (optional)

4.1 Hardware Requirements

You will need the following equipment to complete the Basic Setup configuration:

- AVA-300
- ATV-300 (optional)
- ATM switch
- UNIX workstation with a FORE Systems' ATM adapter card
- VCR or similar video/audio source to generate input to the AVA-300
- TV or similar monitor to display output from the ATV-300 (optional)
- Appropriate ATM cables (e.g., fiber optic cables)
- RCA video and audio cables (shielded, with gold-plated tips recommended)

4.1.1 Workstation Requirements

Your UNIX workstation must contain the proper FORE Systems' ATM adapter card to be connected to the switch. Table 4.1 lists the UNIX operating systems supported by the *StreamRunner* AVA / ATV platform and their corresponding FORE Systems' ATM adapter cards.

Table 4.1 - UNIX Operating Systems and Corresponding Adapter Cards

UNIX Operating Systems	FORE Systems ATM Adapter Card(s)
Solaris 2.5 and 2.6	SBA-200, SBA-200E, or PCA-200 EUX
IRIX 6.3	VMA-200E, GIA-200E, or ESA-200E (v6.2); PCA-200 EUX (v6.3)

As stated at the beginning of this chapter, the Basic Setup configuration assumes you have basic ATM connectivity using UNI 3.0 or UNI 3.1 signalling. If not, please consult the appropriate User's Manual for switch and / or adapter configuration.

NOTE

SVA 5.0 requires *ForeThought* 4.3 UNIX device driver software or greater.

4.2 Installing the SVA Software

After you have set up your hardware with the appropriate adapter card, the next step is to install the SVA software. The SVA software contains programs to operate both the AVA-300 and ATV-300. Also included is a sample stream configuration file. The software is shipped as a compressed tar file that needs to be uncompressed and extracted.

To install the SVA software, do the following:

1. Go to the directory where you wish to install the software. In these examples your HOME directory is assumed. Type one of the following to get to your HOME directory:

   `cd $HOME` `or` **`cd`**

2. Copy the distribution file into your HOME directory.
3. Uncompress the file using the following command:

   uncompress *<your operating system type>***_sva5.0.0.tar.Z**

4. Extract the file contents using the following command:

   tar xvf *<your operating system type>***_sva5.0.0.tar**

NOTE Do not install the release over a previous copy of the SVA distribution. Move or delete any old release prior to performing this step. Be careful not to delete any old manager configuration files that you may want to re-use.

The system creates a directory called `SVA-5.0.0`. The following sub-directories, containing the various files that make up the SVA software distribution, are created in this directory:

./etc	Manager and utility binaries
./config	Manager configuration files
./bin	Application binaries and support files
./lib	SVA (shared) libraries
./man	On-line documentation
./html	HTML files
./plugin	SVA Netscape Plug-in
./firmware	ATV-300 firmware release

You have successfully installed the SVA software.

4.2.1 Environment Variables

Now that you have successfully installed the software, the next step is to set some environment variables to allow you to access the installed binaries and on-line documentation.

To set environment variables, extend your PATH environment variable by typing the following if you are using Bourne shell syntax:

```
prompt$ PATH=$PATH:$HOME/SVA-5.0.0/bin; export PATH
```

or by typing the following if you are using C-shell syntax:

```
prompt% setenv PATH "$PATH":$HOME/SVA-5.0.0/bin
```

Once your PATH environment is set correctly, you can invoke SVA applications. For example, to display the manual page for the avareset command, type the following:

```
svaman avareset
```

The avareset manual page is displayed.

The svarun command is a wrapper script for invoking SVA applications. It automatically sets up any further environment variables that are needed by the SVA software.

NOTE If you would rather use the SVA commands directly instead of the wrapper scripts, refer to Appendix B for details.

NOTE If you know how to edit your login script, you may want to permanently add the SVA bin to your path.

4.3 Configuring the Hardware

The Basic Setup configuration will enable you to send a video image over an ATM network for display on a workstation/PC and a TV. Figure 4.1 illustrates the components you will need and their general positions in the network.

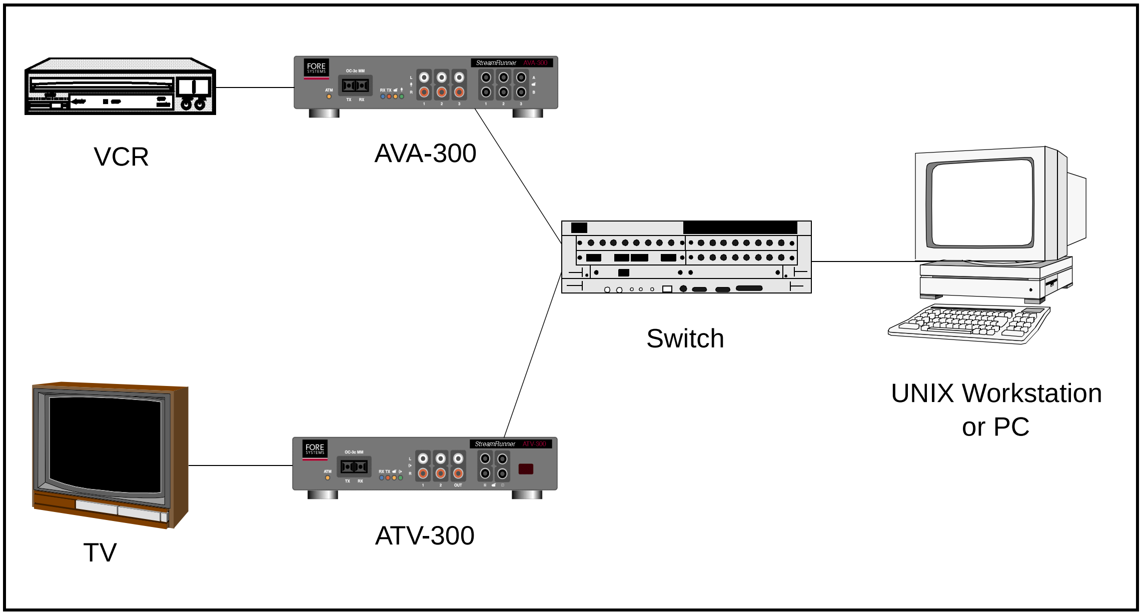

Figure 4.1 - Basic Setup Configuration

The goal of this configuration is to send an analog signal from a VCR tape to the AVA-300. The AVA-300 converts the analog signal to digital and sends it to the ATM switch as ATM cells. The switch in turn sends the cells over a Switched Virtual Circuit (SVC) to the workstation/PC. The SVA software on the workstation/PC decodes and then displays the incoming stream. Additionally, you may also convert the digital signal back to analog through an ATV-300 for display onto the TV. Connecting the hardware consists of five steps:

- Connecting the workstation/PC to the switch
- Connecting the VCR to the AVA-300
- Connecting the AVA-300 to the switch
- Connecting the ATV-300 to the switch (optional)
- Connecting the TV to the ATV-300 (optional)

4.3.1 Connecting the VCR to the AVA-300

To connect the VCR to the AVA-300, perform the following steps:

1. Power up the VCR and AVA-300.

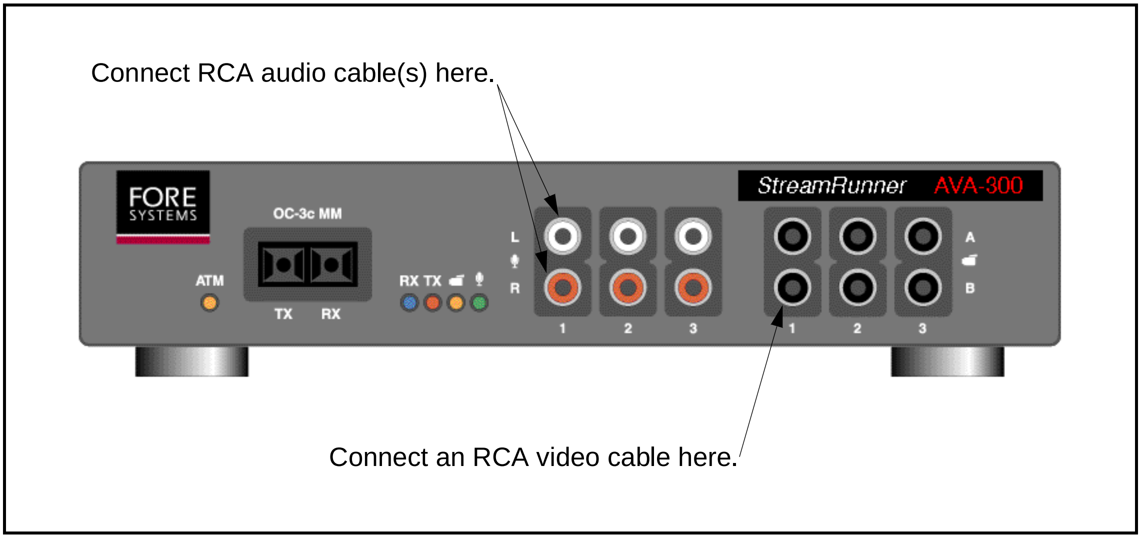

Figure 4.2 - AVA-300 Video and Audio Cable-to-Port Connection Detail

2. Connect one end of an RCA cable to the `1B` video port on the front panel of the AVA-300. Refer to Figure 4.2 for AVA-300 video/audio port locations.
3. Connect the other end of the RCA cable to the video out port on the rear panel of the VCR.

NOTE

`1B` is the default video configuration port.

4. If you have a stereo VCR, connect one end of an RCA cable to the `1L` audio port on the front panel of the AVA-300 and connect the other end to the Left audio port on the rear panel of the VCR. Connect one end of another RCA cable to the `1R` audio port on the front panel of the AVA-300 and connect the other end to the Right audio port on the rear panel of the VCR.

For non-stereo VCRs, you can use either the `1L` or `1R` audio port on the AVA-300.

NOTE

If you are not sure where to connect cables on your VCR or on any other video/audio source you are using in the Basic Setup (e.g., discrepancies in port labelling from different manufacturers), please consult the User's Manual that came with the source equipment for clarification.

The VCR is successfully connected to the AVA-300.

4.3.2 Connecting the AVA-300 to the Switch

To connect the AVA-300 to the switch, perform the following steps:

1. Connect one end of a fiber optic cable to the appropriate pair of OC3 ports on the front panel of the switch. These are usually labeled Rx (receive) and Tx (transmit).

NOTE

Some later examples in this User's Manual assume that the AVA-300 is attached to port `1A1` on switch `orion`. Also assumed is the use of fiber optic cables.

2. Connect the other end of the fiber optic cable to the pair of `OC3` ports on the front panel of the AVA-300.

NOTE

Each connection on the AVA-300 must match the corresponding connection on the switch and vice versa. When you achieve proper connectivity, the `ATM` LED on the AVA-300 changes from red to green, and the red and green receive and transmit LEDs on the switch extinguish. If the `ATM` LED on the AVA-300 remains red or the red and green receive and transmit LEDs on the switch port are illuminated, your equipment is not properly connected. Switch the fiber optic cable pair's positions on either piece of equipment to achieve proper connectivity.

The AVA-300 is successfully connected to the switch.

4.4 User-Directed SPVCs

Previous releases of SVA software relied on the configuration of Permanent Virtual Circuits (PVCs) between AVA-300s, ATV-300s, and the controlling management software. The PVC configuration process can be time-consuming, error-prone, and somewhat inflexible during network re-configuration.

The SVA-5.0 software includes a feature in which the management code running on the workstation or PC is able to contact a remote AVA-300 or ATV-300 by knowing the IP address of the remote switch and the number of the port that it is connected to. The SVA software uses a *ForeThought* facility known as User-Directed Soft Permanent Virtual Circuits (SPVCs). This facility allows client applications using a native ATM API to establish an SVC to a particular switch port on the network without involving the remote attached device in any signalling interaction.

Many of the SVA commands have a `-device` option that specifies the remote AVA-300 or ATV-300. The full syntax allows you to specify the IP address of the destination switch and the port to which the device is attached. The IP address may either be in dot notation form such as `169.144.68.11`, or may be the associated domain name.

The SVA software uses this IP address to query, using SNMP, the NSAP prefix of the remote switch. Therefore, there must be IP connectivity between the client machine and the target switch; in addition, the client must be able to query the remote MIB. The switch port is specified using the standard FORE Systems notation (`1a1, 1a2,` etc.,) which is detailed in the appropriate *ForeRunner* switch documentation.

NOTE In order to use the User-Directed SPVC facility, the switch to which the remote AVA-300 and/or ATV-300 is attached must be running *ForeThought* 4.1.x software or later.

NOTE You may use non-FORE Systems switches between the client and the remote *ForeRunner* switch to which the target device is attached, as long as they transparently pass through the SPVC Information Element component of the connection establishment request.

NOTE You may still use standard PVC configurations as supported in earlier SVA software versions. For more information, please refer to the appropriate Appendices and the Manual and Reference pages located in the back of this User's Manual.

4.5 Verifying Proper AVA-300 Setup

Once you have configured the hardware, installed the software, and set the environment variables, you should verify that you have performed these steps properly. You verify the proper setup using the `avareset` command.

To verify proper setup, use the `avareset` command as follows:

```
svarun avareset -device orion:1A1
```

NOTE Since you have already set the PATH in Section 4.2.1 or Section 5.2.1, you are not required to type in the full path name at this point.

NOTE If you do not wish to use the User-Directed SPVC facility, refer to Appendix C for details on using `avareset` with a PVC control channel.

If set up properly to this point, the system displays version information similar to the following:

```
Hardware:     300.13 (CCube Rev. E)
Firmware:     4.3
Serial No:    97000000
Interface:    ATM 155 Mbps
Release:      ForeThought 5.0.0 (SVA sva50a2)
```

NOTE If your firmware version is less than 4.3, then it is strongly recommended that you consider upgrading your firmware by contacting FORE Systems Technical Support for a *StreamRunner* AVA-300 PROM Upgrade Kit.

If not set up properly, the system displays the following error message:

```
avareset: failed to contact AVA ... retry
```

NOTE If you receive this error message, double-check your connections and/or your configuration syntax. Also make sure that no managers are running. Refer to Section 4.6 for a description of managers.

4.5.1 Verifying Video Environment

The AVA-300 and ATV-300 devices support either PAL (Phase Alternate Lines) or NTSC (National Television Systems Committee) standard video. The standard you use typically depends on your location: NTSC is used primarily in the United States and Japan whereas PAL is used primarily in the United Kingdom and Europe. A more detailed discussion on video standards may be found in Section 7.5.4.

The SVA software uses the local time zone of the computer on which it is running to make an informed choice as to the default video standard to use. In order to ascertain the default video standard selected, type the following:

`svarun svavideo`

The system displays on of the following messages:

```
The default video standard is PAL
```

or:

```
The default video standard is NTSC
```

If the video standard is incorrect for your environment, then a number of alternatives exist:

1. The time zone on your computer may be set incorrectly. If this is the case, reset the time zone accordingly.
2. Set the `VSTANDARD` environment variable to either `PAL` or `NTSC`. This variable is inspected on SVA command startup and will override the default based on the local time zone.
3. Some SVA commands, such as `avaconfig`, take the video standard to use as a command line argument, such as `-pal` or `-ntsc`. Using this method overrides both the time-zone-based default and any `VSTANDARD` setting.

4.6 Setting Up a Trader and AVA-300 Manager

After verifying the proper setup, your next steps involve:

- Starting a trader
- Registering the AVA-300's manager with the trader

The `trader` is an application that lists and keeps track of the available managers. Therefore, the `svamgr` software representing an AVA-300 needs to register with the trader.

The `manager` is an SVA software application that is responsible for setting up audio and video streams. The application is called `svamgr`. You may use a program called `svarun` to invoke the `svamgr`.

This example shows one AVA-300, and, therefore, one manager being registered with one trader. In actual network applications, one trader running on a single network server is sufficient for most installations. However, it is possible to have many traders available for managers to register with.

4.6.1 Starting a Trader

You must now start a trader to provide a place for your manager to register. A video network could be comprised of several (and at least one) traders.

To start a trader, open a window on the workstation / PC and type the following:

```
svarun trader
```

NOTE

Since a trader must run continuously, it must either run in its own window or in the background.

The following message is displayed to let you know that the `trader` application is running:

```
(Date and time): trader: started on host ”phoenix”
```

You have started a trader.

4.6.2 Registering the AVA-300's Manager with the Trader

NOTE

If your AVA-300 firmware version is less than 4.3, then refer to Appendix F for details on starting the AVA-300 manager.

NOTE

If you are not using the User-Directed SPVC facility, then refer to Appendix E for details on starting the AVA-300 manager.

Once you have started the trader, you need to start the AVA-300 manager so it registers with the trader. To register a manager, open a window on the workstation / PC and type the following:

```
svarun -name ava1 -device orion:1A1
```

NOTE

SVA 5.0 defaults to run managers with UNI 3.0 unless otherwise instructed. If you are restricted to using UNI 3.1 signalling, you must type the following command:

```
svarun -name ava1 -device orion:1A1 -uni31
```

The following message lets you know that the `manager` application is running:

```
(Date and time): svamgr: started on host ″phoenix″
```

In addition, a list of available video and audio streams follows the message.

You have started the AVA-300 manager and registered it with the trader.

4.7 Running svc-rtds

Once your trader and manager are running, you can use the SVC Real-Time Display Software (`svc-rtds`) application to create, display, and edit video and audio streams on your workstation / PC. `svc-rtds` needs to know the IP address or hostname of a workstation on which a trader was started. If no trader is specified, `svc-rtds` assumes there is one running on the local machine.

To run `svc-rtds`, do the following:

1. Open a window on the workstation / PC.
2. At the prompt, type:

   `svc-rtds -edit`

   The `AVA/ATV Manager Browser`, shown in Figure 4.3, is displayed.

Figure 4.3 - AVA / ATV Manager Browser Window

   The `AVA/ATV Manager Browser` is a GUI containing a list of managers. In the Basic Setup, the list is composed of the one manager you named, started, and registered with the trader earlier. (This example calls it `ava1.`)

3. Double-click on `ava1` or highlight it with a single-click and then click on the `Access Manager` button.

The AVA/ATV Manager Window, similar to that shown in Figure 4.4, is displayed.

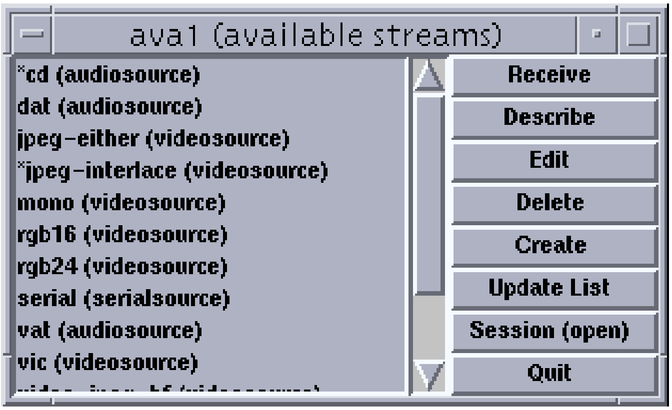

Figure 4.4 - AVA / ATV Manager Window

This window displays the audio and video streams available to designated manager `ava1`. In the Basic Setup, these are from the default stream file (`svadefaults`) in the directory specified by the `-configdir` parameter which is automatically set by the `svarun` command.

4. With a tape running in your VCR, double-click on the video stream named `mono` or highlight it with a single click and then click on `Receive` to start the stream.
5. Double-click on the audio stream named `cd` or highlight it with a single click and then click on `Receive` to start the stream.

NOTE

The streams you have joined are defined to be easily decoded and displayed on most workstations / PCs. Depending on your machine, you will be able to experiment with other stream definitions to achieve better stream quality.

System resources may limit your ability to display some video streams. For example, a workstation lacking decompression hardware support may be unable to decode and display a full-frame rate compressed video stream. Refer to Chapter 10 to find out more about resolving video quality issues.

You have now completed the AVA-300 Basic Setup configuration. This provided the basic information required to send video and audio over an ATM network. The rest of this User's Manual will provide you with more in-depth information to explore and use the *StreamRunner* AVA / ATV platform to its fullest advantage. Refer to the next section for information on adding an ATV-300.

4.8 Connecting an ATV-300 and TV to the Network

Your ATM network may include one or more ATV-300s. If your setup requires the additional flexibility of decoding digital video and audio streams back to a conventional analog signal, perform the tasks in the following sections.

4.8.1 Connecting the ATV-300 to the Switch

To connect the ATV-300 to the switch, do the following:

1. Power up the ATV-300.
2. Connect one end of a fiber optic cable to the appropriate pair of OC3 ports on the front panel of the switch. These are usually labeled Rx (receive) and Tx (transmit).

NOTE Some later examples in this User's Manual assume that the ATV-300 is attached to port `1B1` on switch `orion`. Also assumed is the use of fiber optic cables.

3. Connect the other end of the fiber optic cable to the pair of `OC3` ports on the front panel of the ATV-300.

NOTE Each connection on the ATV-300 must match the corresponding connection on the switch and vice versa. When you achieve proper connectivity, the `ATM` LED on the ATV-300 changes from red to green, and the red and green receive and transmit LEDs on the switch extinguish. If the `ATM` LED on the ATV-300 remains red or the red and green receive and transmit LEDs on the switch port are illuminated, your equipment is not properly connected. Switch the fiber optic cable pair's positions on either piece of equipment to achieve proper connectivity.

The ATV-300 is successfully connected to the switch.

4.8.2 Connecting the ATV-300 to the TV

To connect the ATV-300 to the TV, do the following:

1. Power up the TV. Refer to Figure 4.5 for ATV-300 video/audio port locations.

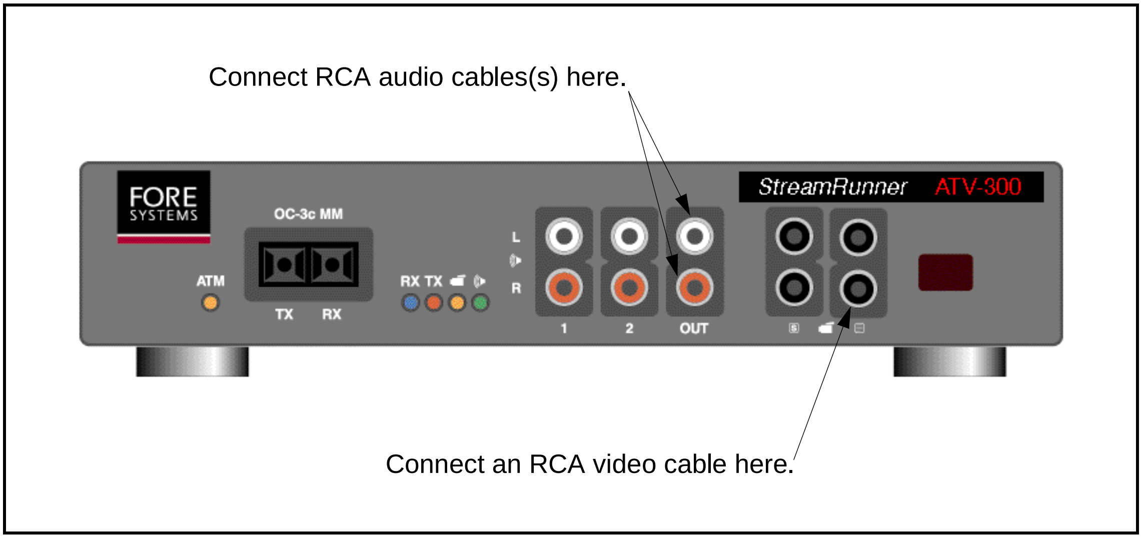

Figure 4.5 - ATV-300 Video and Audio Cable-to-Port Connection Detail

2. Connect one end of an RCA cable to the `composite video port` (identified by the ⊟ symbol) on the front panel of the ATV-300.
3. Connect the other end of the RCA cable to the video input port on the rear panel of the TV.
4. If you have a stereo TV, connect one end of an RCA cable to the `L out` audio port on the front panel of the ATV-300 and connect the other end to the Left In audio port on the rear panel of the TV. Connect one end of another RCA cable to the `R out` audio port on the front panel of the ATV-300 and connect the other end to the Right In audio port on the rear panel of the TV.

   For non-stereo TVs, you can use either the `L out` or `R out` audio port on the ATV-300.

   The ATV-300 is successfully connected to the TV.

4.9 Verifying Proper ATV-300 Setup

Once you have configured the hardware, you should verify that your ATV-300 is properly set up, as you did in Section 4.5 for the AVA-300. You verify the proper setup using the `atvreset` command.

To verify proper setup, use the `atvreset` command as follows:

```
svarun atvreset -device orion:1B1
```

NOTE If you do not wish to use the User-Directed SPVC facility, refer to Appendix C for details on using PVC control channels.

If the control channel is configured properly, the system displays version information similiar to the following:

```
Hardware:     300.5 (CCube Rev. E)
Firmware:     300.18
Serial No:    97000000
Interface:    ATM 155 Mbps
Release:      ForeThought 5.0.0 (SVA sva50a2)
```

NOTE If your firmware version is less than 300.18, then refer to Appendix D.

If the control channel is not configured properly, the system displays the following error message:

```
atvreset: failed to contact ATV ... retry
```

NOTE If you receive this error message, double-check your connections and/or your configuration syntax.

4.10 Setting Up the ATV-300 Manager

After testing the control channels and determining that they are properly set up, your next step involves registering the ATV-300's manager with the trader.

4.10.1 Registering the ATV-300's Manager with the Trader

NOTE If you are not using the User-Directed SPVC facility, then refer to Appendix E for details on starting the ATV-300 manager.

You must start the ATV-300 manager so it registers with the trader that you configured in Section 4.6.1. To register a manager, open a window on the workstation/PC and type the following:

```
svarun -name atv1 -device orion:1B1
```

NOTE SVA 5.0 defaults to run managers with UNI 3.0 unless otherwise instructed. If you are restricted to using UNI 3.1 signalling, you must type the following command:

```
svarun -name atv1 -device orion:1B1 -uni31
```

The following message lets you know that the `manager` application is running:

```
(Date and time): svamgr: started on host ″phoenix″
```

You have started the ATV-300 manager and registered it with the trader.

NOTE

Once you have started the ATV-300 manager and registered it with the trader, its name (`atv1` in this example) is displayed in the `AVA/ATV Manager Browser` list above the AVA-300 manager named earlier (`ava1`), as shown in Figure 4.6.

Figure 4.6 - AVA / ATV Manager Browser Window

4.11 Running svapatch

In order to get video and audio onto the TV by way of the ATV-300, an SVC must be established between the AVA-300 and the ATV-300. You do this using `svapatch`.

To connect a video and audio stream to the ATV-300 using `svapatch`, type the following on your workstation/PC:

```
svarun svapatch -from ava1 -to atv1 jpeg-either cd
```

The video image and associated audio sound from the VCR is displayed and heard on the TV.

NOTE Both the AVA-300's and ATV-300's managers must be started and registered for `svapatch` to be successful (as verified by the `AVA/ATV Manager Browser` window in Figure 4.6).

NOTE The video quality displayed by the `jpeg-either` stream does not represent the best video quality you can achieve.

You have added an ATV-300 to the Basic Setup configuration and have completed all of the tasks in this chapter. Refer to the other chapters in this User's Manual to acquire more in-depth information on the hardware, software, and system configuration possibilities available to you.

CHAPTER 5 Windows NT/95 Basic Setup

This chapter is a step-by-step guide to get you started with your *StreamRunner* AVA/ATV hardware and SVA software system over an ATM network using a Windows NT or Windows 95 platform. Included are an overview of the system components and how to configure them, and how to install, configure, and use the software. This chapter assumes:

- A working knowledge of Windows NT and/or Windows 95.
- A basic knowledge of ATM networking concepts
- ATM connectivity using UNI 3.0 or UNI 3.1 signalling

The Basic Setup involves the following steps, each of which is detailed in this chapter:

- Installing the SVA Software
- Configuring the Hardware
- Verifying Proper Setup
- Setting up a Trader and Manager
- Running the Real-Time Display Software (`svc-rtds`)
- Running `svapatch` (optional)

5.1 Hardware Requirements

You will need the following equipment to complete the Basic Setup configuration:

- AVA-300
- ATV-300 (optional)
- ATM switch
- PC with a FORE Systems' ATM adapter card
- VCR or similar video/audio source to generate input to the AVA-300
- TV or similar monitor to display output from the ATV-300 (optional)
- Appropriate ATM cables (e.g., fiber optic cables
- RCA video and audio cables (shielded, with gold-plated tips recommended)

5.1.1 PC Requirements

Your PC must contain the proper FORE Systems' ATM adapter card to be connected to the switch. Table 5.1 lists the PC operating systems supported by the *StreamRunner* AVA / ATV platform and their corresponding FORE Systems' ATM adapter cards.

Table 5.1 - PC Operating Systems and Corresponding Adapter Cards

PC Operating Systems	FORE Systems ATM Adapter Cards
Windows NT 4.0 (SP3)	PCA-200E or LE-155
Windows 95	PCA-200E or LE-155

NOTE

To work with SVA 5.0 properly, Windows NT 4.0 must be running Service Pack 3 (SP3). If you do not have SP3, it may be downloaded from Microsoft free-of-charge.

NOTE

Winsock2.SDK files must be installed on your PC to run the Windows 95 software.

As stated at the beginning of this chapter, the Windows Basic Setup configuration assumes you have basic ATM connectivity using UNI 3.0 or UNI 3.1 signalling. If not, please consult the appropriate User's Manual for switch and / or adapter configuration.

5.1.2 Windows 95 Support

Windows 95 does not support "servers" such as `svamgr` and `trader`. To use the full suite of SVA software, you should upgrade to Windows NT 4.0.

NOTE

The Basic Setup cannot be performed using Windows 95.

5.2 Installing the SVA Software

After you have set up your hardware and confirmed that you have connectivity, the next step is to install the SVA software. The SVA software contains programs to operate both the AVA-300 and ATV-300. Also included is a sample stream configuration file. The software is shipped with the hardware on CD-ROM, and is also available over the Internet

To install the SVA software, do the following:

1. Insert the CD into your CD-ROM drive or download the software from the Internet. A list of all the files contained in the SVA 5.0 software is displayed.
2. Double click on `SETUP.EXE`. The *ForeThought* `SVA-5.0 Setup Screen` is displayed.

NOTE Do not install this release over a previous copy of the SVA distribution. Move or delete any old release prior to performing this step. Please refer to Section 5.3. Be careful not to delete any old manager configuration files that you may want to re-use. It is also advisable to exit from any other Windows-based applications you may be running.

3. Follow the on-screen setup instructions that are displayed. Read the information contained in each window and click on the `Next>` button to advance through the setup sequence.

NOTE The setup installs the SVA 5.0 software in the default destination folder `C:\Program Files\FORE Systems, Inc.\ForeThought SVA 5.0`. Click on `Browse...` to select another folder if you choose to do so.

NOTE `NRLTRADERS` is a list of host machines that are running traders. On the `NRLTRADERS Screen`, you are prompted to enter the hostname(s) of the machine(s) that are running traders and managers that you may wish to access. Consult your system administrator if you are not sure of the appropriate hostnames. In addition, traders and managers are not supported on machines running Windows 95; host machines must be running either Windows NT or UNIX.

NOTE Windows 95 does not automatically provide a path for the HOME directory. Therefore, when prompted by the `HOME Screen`, provide the system with the location of your HOME directory.

4. Complete the software installation by clicking `Finish` in the `Setup Complete Screen`.

   You have successfully installed the SVA software.

NOTE

If you are running as the administrative user on Windows NT, you have the option of making the installation available to all users (recommended); other users may only install the software for themselves.

In the current version of the installer, you cannot install the software more than once on a machine; if user A installs it, user B cannot install or remove it, or view it from their `Start Menu` or on their PATH.

Therefore, the administrative user is strongly advised to install the software on the machine before another user does, thus allowing all users to see it.

5.2.1 Environment Variables

`SETUP.EXE` automatically sets the environment variables to their proper settings during the installation process. While the defaults are suitable for most applications, you may edit the environment variables to match your system requirements.

5.2.1.1 Setting Environment Variables in Windows NT

The environment variables may be modified at the `System Variables Window`. To access the `System Variables Window` in order to set the environment variables in Windows NT, do the following:

1. Click on `Start`, then on `Settings`, then on `Control Panel`, and finally on `System`.
2. In the `System Properties Window`, click on `Environment`. The `System Variables Window`, shown in Figure 5.1, is displayed.

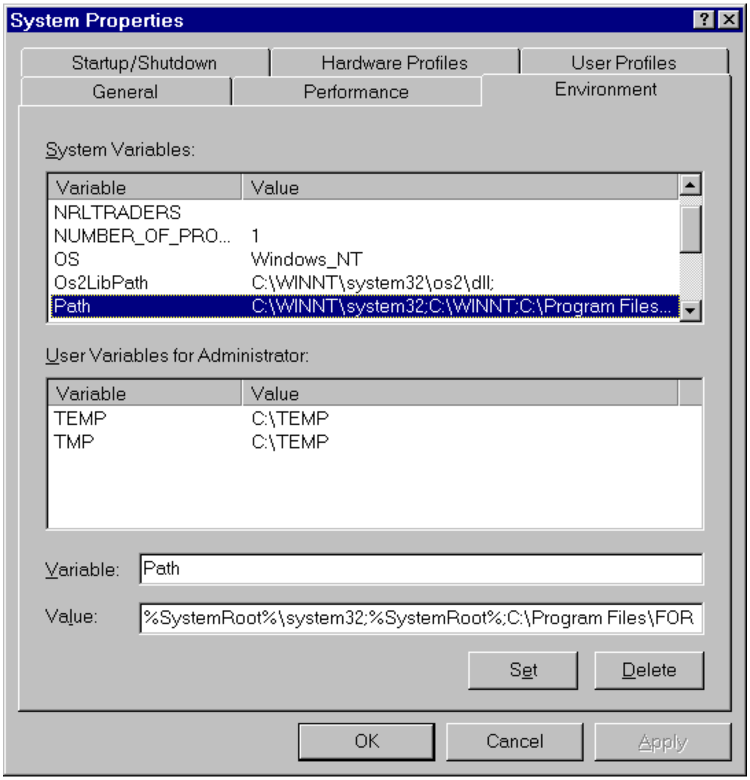

Figure 5.1 - System Variables Window

3. Modify either the `PATH` or `NRLTRADERS` settings by clicking on either variable and editing its value when it is displayed in the `Value` field.
4. Click `Set` to set the modified variable value.
5. Click `OK` when you have made all of your required modifications.

5.2.1.2 Setting Environment Variables in Windows 95

The environment variables may be modified through the `AUTOEXEC.BAT` file located in the software distribution. To set the environment variables in Windows 95, do the following:

1. Open a DOS shell and access the `AUTOEXEC.BAT` file by typing the following at the C:\ prompt:

   `EDIT AUTOEXEC.BAT`

2. Type your desired path as follows:

   `PATH=%PATH%;`*`<drive>`***`\`***`<path>`*

   where *`<drive>`* is the drive on which you wish to install and *`<path>`* is the path to the “bin” directory of the SVA installation.

NOTE The system automatically sets the following path at setup:

`c:\Program Files\ FORE Systems, Inc.\FORETHOUGHT5.0\SVA5.0\BIN`

NOTE Windows 95 users are advised to use the short (8.3) filename format without spaces. Some installers corrupt the `autoexec.bat` file if you put spaces in the PATH definition

5.3 Removing the SVA Software

It is important that you do not move or delete any files (except manager configuration files) in the SVA software distribution from their installed location, except by the recommended method. Therefore, to properly remove the SVA software distribution, do the following:

1. Click on `Start`, then on `Settings`, then on `Control Panel`, and finally on `Add/Remove Programs`.
2. Double-click on `ForeThought SVA-5.0`.

   The software distribution is removed in the proper manner.

NOTE

If you receive the message:

`"Internal error, unable to load or call external DLL. Please contact your vendor for more information,"`

attempt a reinstall. The installer will detect the problem and correct it. After doing this, perform the steps detailed above again.

On the rare occasion that these procedures do not properly remove the SVA software distribution, please refer to Appendix H for further instructions.

5.4 Configuring the Hardware

Once your environment variables have been set, the next step is to configure the hardware. Turn to Section 4.3 to complete the Basic Setup.

CHAPTER 6 SVA Control

This chapter introduces you to SVA Control, the Windows NT graphical user interface for implementing SVA program tasks including managers, traders, and patches. Although these tasks can also be started from a command prompt within a DOS shell, as described in Chapter 5, SVA Control provides an alternate tool that may be favored by Windows operating system users.

6.1 Accessing SVA Control

To access SVA Control, do the following:

1. Click on `Start`, then on `Programs`, then on `ForeThought SVA-5.0`, and finally on `SVA Control`.

   The `SVA Control Main Window`, shown in Figure 6.1, is displayed.

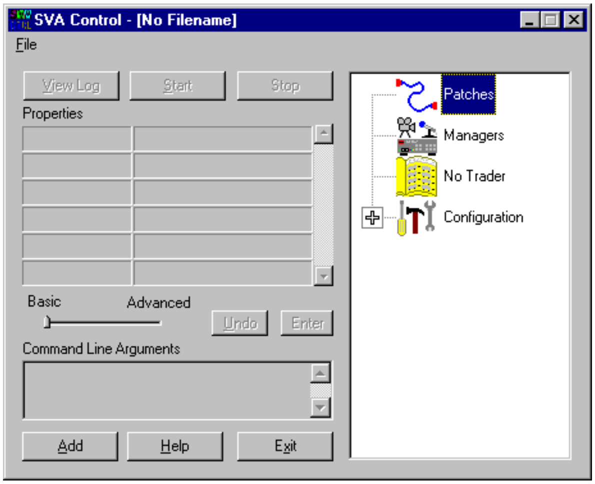

Figure 6.1 - SVA Control Main Window

NOTE In addition to the `SVA Control Main Window`, a console window where logs are displayed, is also displayed.

The window on the right displays a tree that shows the tasks that are or can be configured. It works like the Windows NT Explorer application; the tree may be expanded or collapsed by clicking on the “plus” or “minus” icons located on the extreme left. If no managers or patches are defined, the tree is empty (with no underlaying branches) and a plus or minus icon is not present. There may be any number of managers and patches, and up to one each trader or patch server.

NOTE Managers and traders are not supported on Windows 95; you must use Windows NT to utilize these program features.

The `Configuration` tools do not represent specific tasks, but instead run for only a few seconds, which is the time required to download the configuration to the device. After doing so, they stop.

The `Properties` dialog to the left is where you enter or modify the parameters of the task you have highlighted in the right. Pressing the `Undo` button returns the highlighted property to its original setting before you made the error.

The `Command Line Arguments` box displays the commands you invoke when performing tasks with SVA Control. Any application’s progress or current status is also displayed in the DOS console window which was opened along with SVA Control.

6.2 SVA Control and Basic Setup

As stated earlier, any tasks you wish to complete with a DOS shell can also be implemented within SVA Control. The following sections detail:

- Starting a Trader
- Registering an AVA-300's Manager
- Registering an ATV-300's Manager
- Patching the AVA-300 and ATV-300

6.2.1 Starting a Trader

To start a trader with SVA Control, do the following:

1. Click on the `Trader` icon in the `SVA Control Main Window`.

   The `No Trader` icon illuminates as shown in Figure 6.2.

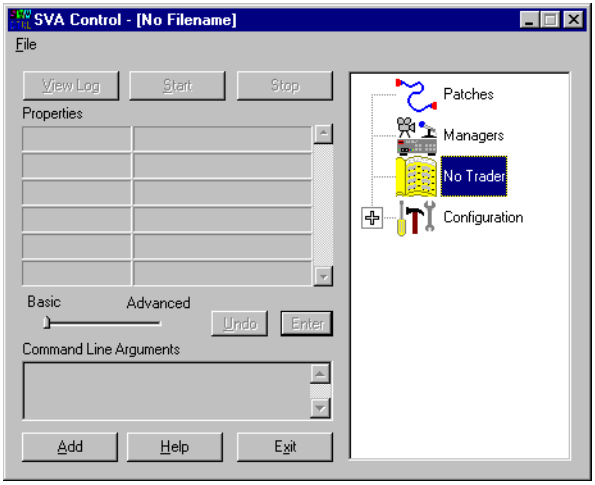

Figure 6.2 - Accessing the Trader Button

2. Click on Add.

The trader is added to the `SVA Control Window`, as reflected in the `Trader` icon being displayed and illuminated, as in Figure 6.3.

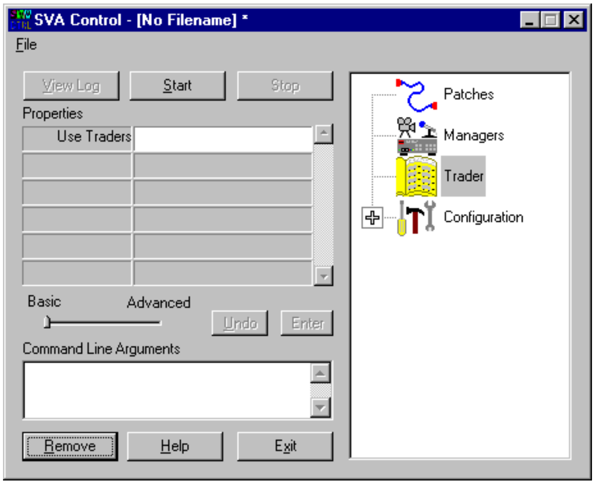

Figure 6.3 - Trader Icon Illumination

In addition, the `Add` button is replaced by the `Remove` button and the `Start` button is illuminated.

3. Click on the `Start` button.

   The Trader is started, and the following message is displayed in the underlaying command prompt window to let you know that the `trader` application is running:

```
(Date and time): trader: started on host "phoenix"
```

You have started a trader.

6.2.2 Registering an AVA-300's Manager

To register the AVA-300's manager, do the following:

1. Click on the `Managers` button.

   The `Managers` icon is illuminated, as shown in Figure 6.4.

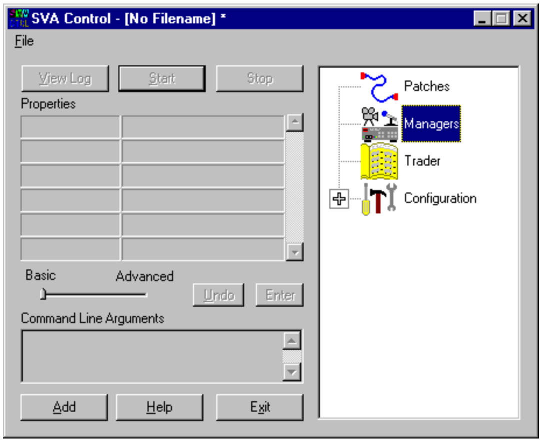

Figure 6.4 - Manager Icon Illumination

2. Click on the Add button.

   The "plus" icon is displayed on the left of the icon tree, as shown in Figure 6.5.

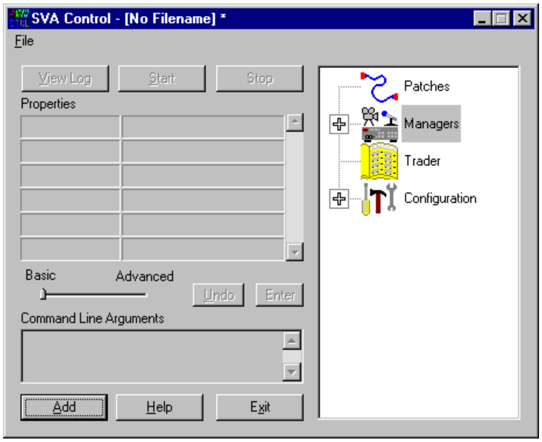

Figure 6.5 - Add Manager Screen

3. Double-click on the `Managers` icon.

   The `Unassigned` icon is displayed underneath, as shown in Figure 6.6.

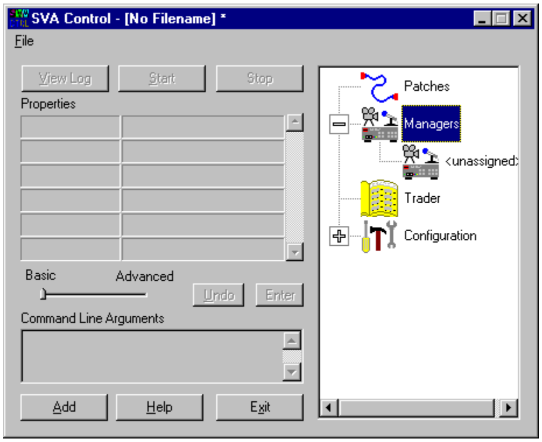

Figure 6.6 - Unassigned Manager Screen

4. Click on the `Unassigned` icon.

   The `Properties` dialog is illuminated, as shown in Figure 6.7.

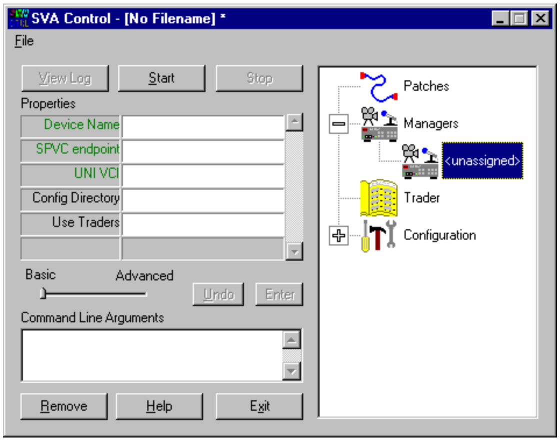

Figure 6.7 - Properties Dialog Screen

5. Click in the `Device Name` field.
   The `Properties` dialog remains illuminated while the rest of the screen is grayed to the background.
6. Type `ava1` in the `Device Name` field, then `Tab` to the `SPVC endpoint` field.
   Type `orion:1A1` in the `SPVC endpoint` field.

   Figure 6.8 illustrates the result of this procedure.

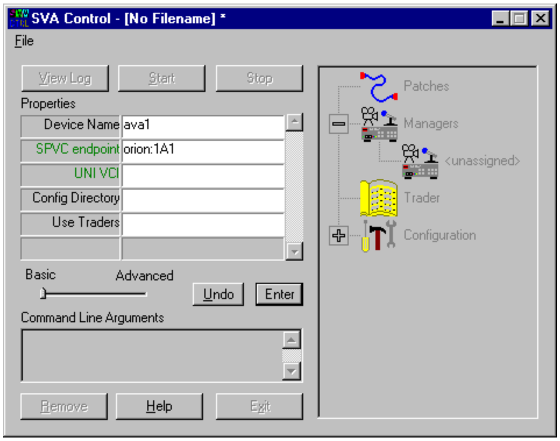

Figure 6.8 - Enter Properties Screen

7. Click on the `Enter` button.
   The screen illuminates and the `ava1` manager is assigned. In addition, the `Command Line Arguments` box displays the command you have just invoked. Figure 6.9 illustrates the result of this procedure.

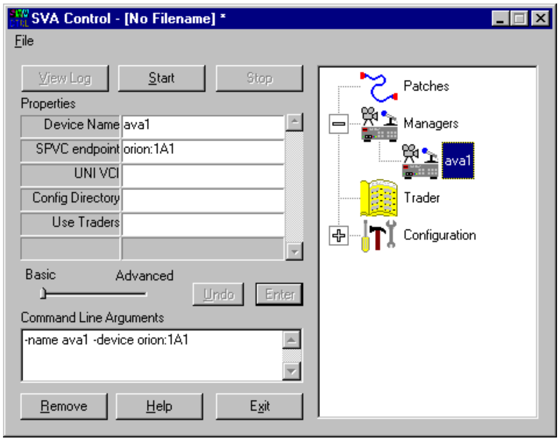

Figure 6.9 - Assign AVA-300 Manager Screen

8. Click on the `Start` button.

   The screen display shown in Figure 6.10 lets you know that the AVA-300 manager application is running. Also, the following message is displayed in the underlying command prompt window to let you know that the `manager` application is running:

```
(Date and time): svamgr: started on host "phoenix"
```

   In addition, a list of available video and audio streams follows the message.

   You have started the manager and registered it with the trader.

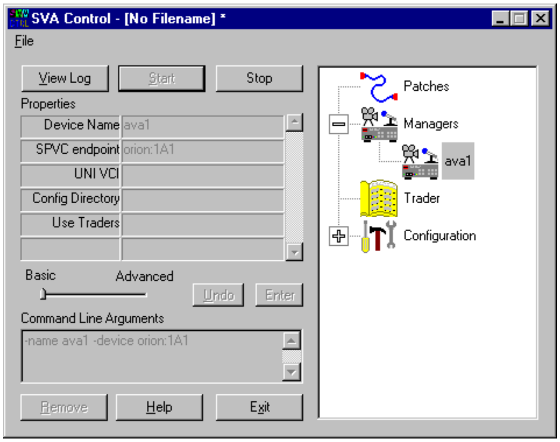

Figure 6.10 - AVA-300 Manager Application Running Screen

6.2.3 Registering an ATV-300's Manager

To register the ATV-300's manager, follow the procedures outlined in the previous section. Be sure to type `atv1` in the `Device Name` field and `orion:1B1` in the `SPVC endpoint` field.

6.2.4 Patching the AVA-300 and ATV-300

Once you register the AVA-300 and ATV-300, the next step is to establish an SVC between the AVA-300 and the ATV-300. To patch the two devices, do the following:

1. Click on the `Patches` button.

   The `Patches` icon is illuminated, as shown in Figure 6.11.

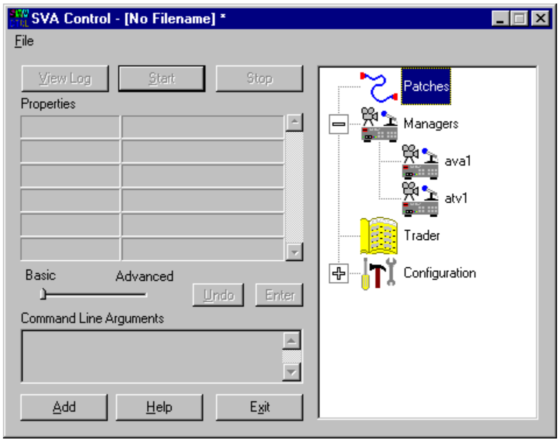

Figure 6.11 - Patches Icon Illumination

2. Click on the Add button.

   The "plus" icon is displayed on the left of the icon tree, as shown in Figure 6.12.

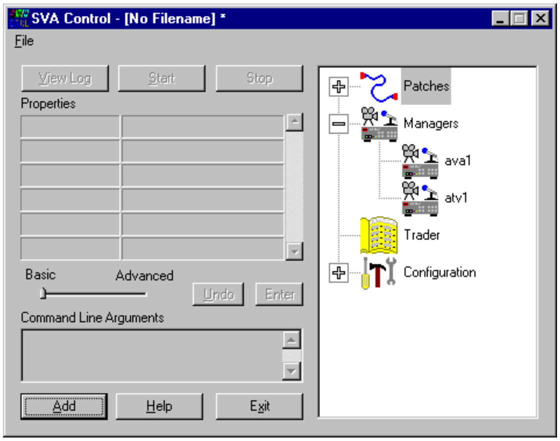

Figure 6.12 - Add Patch Screen

3. Double-click on the `Patches` icon.

   The `Unassigned` icon is displayed underneath, as shown in Figure 6.13.

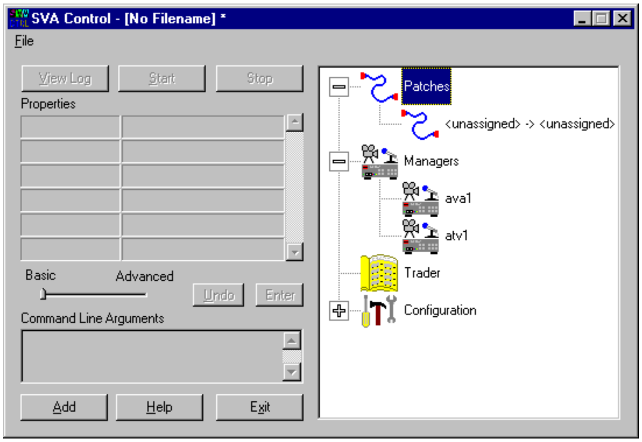

Figure 6.13 - Unassigned Patches Screen

4. Click on the `Unassigned` icon.

   The `Properties` dialog is illuminated, as shown in Figure 6.14.

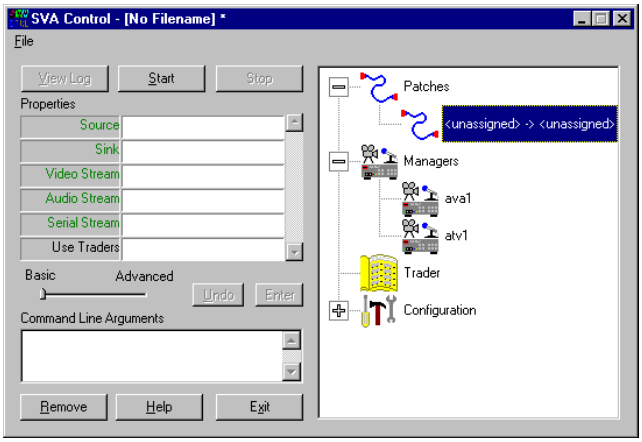

Figure 6.14 - Properties Dialog Screen

5. Click in the `Source` field. Type `ava1` in the `Source` field, `atv1` in the `Sink` field, `jpeg-interlace` in the `Video Stream` field, and `cd` in the `Audio Stream` field.

   Figure 6.15 illustrates this procedure.

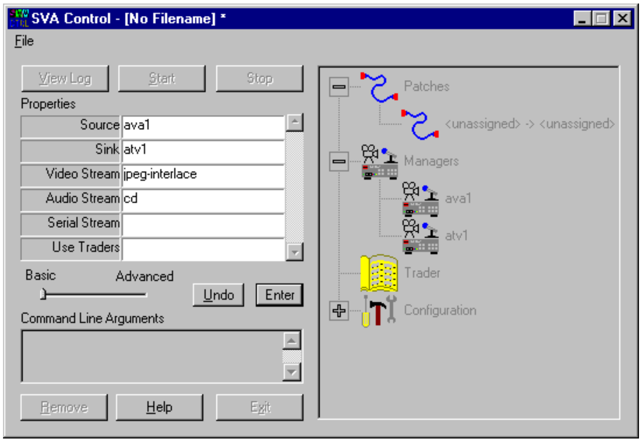

Figure 6.15 - Enter Properties Screen

SVA Control

6. Click on the `Enter` button.

   The screen illuminates and the `ava1-to-atv1` patch is assigned. In addition, the `Command Line Argument` box displays the command you have just invoked. Figure 6.16 illustrates the result of this procedure.

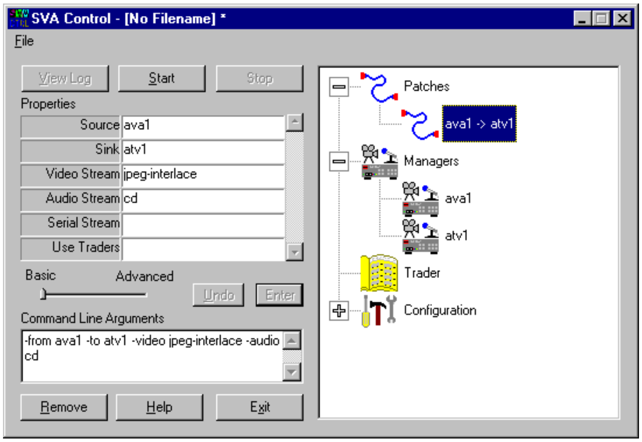

Figure 6.16 - Assign Patch Screen

7. Click on the `Start` button.

   The screen display shown in Figure 6.17 lets you know that the `patch` application is running. Also, the video image and associated audio sound from the VCR is displayed and heard on the TV.

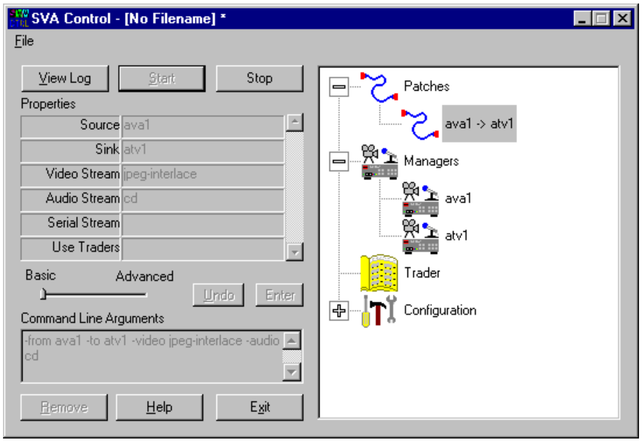

Figure 6.17 - Patch Running Screen

You have completed the Basic Setup configuration using the SVA Control interface.

6.3 Load/Save File

After completing the Basic Setup, you can perform a number of operations which allow you save and open the configuration file you have just created.

6.3.1 Saving the Basic Setup File Configuration

To save the Basic Setup file configuration, do the following:

1. Go to the `File` menu and select `Save As`.
2. Type `Basic` in the `Filename` dialog box and press `Ok`.

   The Basic Setup configuration is saved to disk.

6.3.2 Creating a New File Configuration

To create a new file configuration, do the following:

1. Go to the `File` menu and select `New`.

   You have a “clean slate” with which you can create a new configuration.

6.3.3 Opening an Existing File Configuration

To open an existing file configuration, do the following:

1. Go to the `File` menu and select `Open`.
2. Use the Explorer-like interface to find a previously-saved file configuration.
3. When you find the file you wish to open, select it and press `Ok`.

   The file configuration loads back from disk.

CHAPTER 7 SVA Software

This chapter discusses the SVA Software distribution, its components, and the overall software system architecture by explaining the underlying concepts and their application.

7.1 The SVA Software Distribution

The SVA Software distribution is stored on CD-ROM (and may also be downloaded from the Internet) and must either be uncompressed or extracted depending upon your operating system. Instructions for doing this are detailed in Chapter 4 for UNIX users and Chapter 5 for Windows users earlier in this manual. The distribution includes the following program files, which are discussed in detail in various parts of this manual:

atmdrivers Shows the ATM driver-specific libraries with which the SVA software release is linked with.

atvconfig Allows you to load video and audio stream configurations to an ATV-300 from a remote workstation.

atvdownload Allows you to load a firmware upgrade remotely onto an ATV-300.

atvmenu Allows you to generate and display menus and submenus on a remote ATV-300 and processes the selected requests.

atvpatch Allows you to instruct a remote ATV-300 to display a menu of AVA-300s to connect to when requested.

atvreset Allows you to reset, test proper connectivity, and verify general ATV-300 configuration information.

avaconfig Allows you to load video and audio stream configurations to an AVA-300 from a remote workstation.

avareset Allows you to reset, test proper connectivity, and verify general AVA-300 configuration information.

ecmconfig Allows you to set and/or clear video and audio streams stored in an External Configuration Module, which is attached to the rear panel of an AVA-300.

killmgr Allows you to remotely terminate manager processes.

listchain Lists AVA-300s and ATV-300s attached to a single switch port.

mgrls Allows you to query managers registered in specified traders.

probepvc Allows you to check the availability of specified Permanent Virtual Circuits (PVCs).

svaman Allows you to find and display the SVA manual pages.

svamgr Allows you to manage remote AVA-300 and ATV-300 devices.

svapatch Allows you to connect video and audio streams remotely from an AVA-300 to sink devices such as the ATV-300 or a workstation monitor.

svarun Provides a wrapper script for SVA applications.

svavers Displays the version information of the SVA software release.

svavideo Reports default video standard.

svc-rtds Provides a GUI for editing, displaying, playing, and saving video and audio streams generated by an AVA-300.

7.2 System Architecture

The SVA software uses a client server architecture to separate the management of devices from their use by applications.

7.2.1 Devices

The AVA-300 and ATV-300 hardware components are sometimes referred to as devices. These components, fully described in their chapters earlier in this manual, are stand-alone devices whose only connection to other computers is through the ATM network. They are managed over the network by device-specific managers, which are described in the next section.

7.2.2 Managers

Managers are SVA software applications that have two primary functions:

- maintain ATM signalling on behalf of the device
- make the device accessible to applications

Managers control all aspects of a device's operation, particularly the interaction with the ATM network to establish Switched Virtual Circuits (SVCs) dynamically for the video and audio streams generated or consumed by the device. Maximum performance is ensured since all video and audio data is communicated directly over the ATM network and does not flow through the manager.

Together, devices and managers constitute a service which provides video and audio streams for use by applications. One such application is `svc-rtds`, which stands for Real-Time Display Software and conveys the fact that `svc-rtds` provides real-time presentation of video and audio streams on your workstation. The interaction between devices, managers, and applications is illustrated in Figure 7.1.

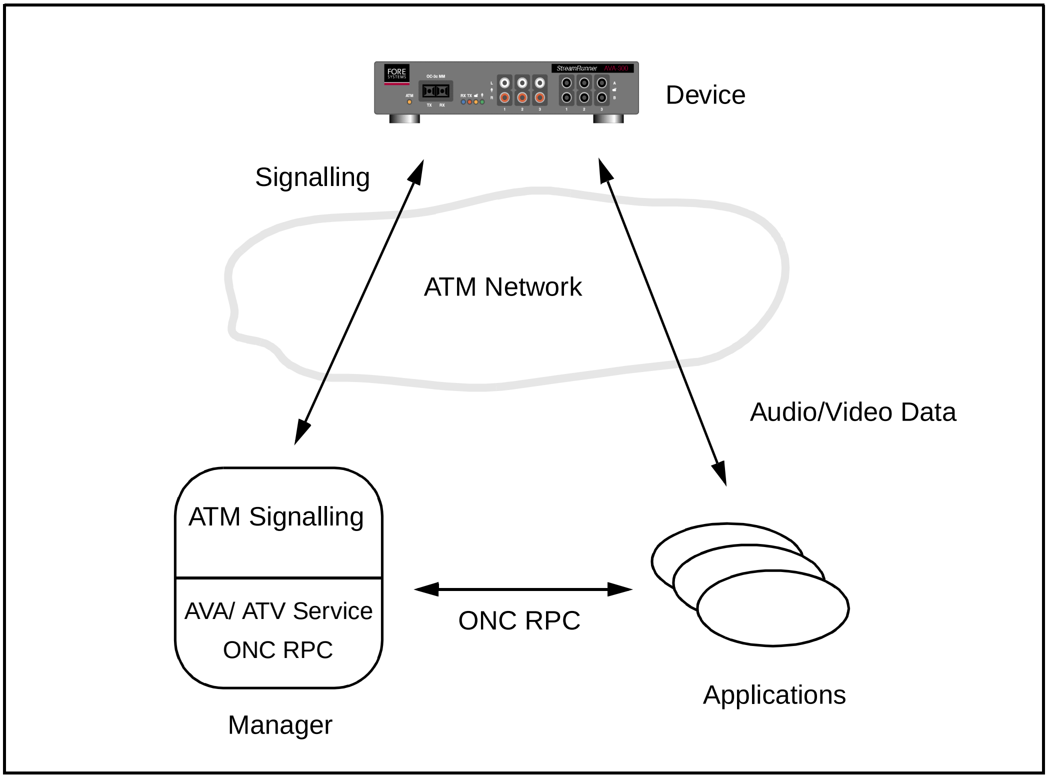

Figure 7.1 - How a Device, Manager, and Applications Interact

Managers also provide an ONC (Sun) Remote Procedure Call (RPC) interface for use by client applications. This interface allows a single device to be accessed by multiple applications and their users simultaneously, and provides a high-level view of the device's video and audio stream configuration and operation.

7.2.3 Traders

Traders provide a distributed database with which managers register their existence and from which applications locate all running managers. In this way, applications do not need to know the location of managers, just the name of the host machine(s) running a trader(s). Figure 7.2 illustrates the interaction between managers, traders, and applications.

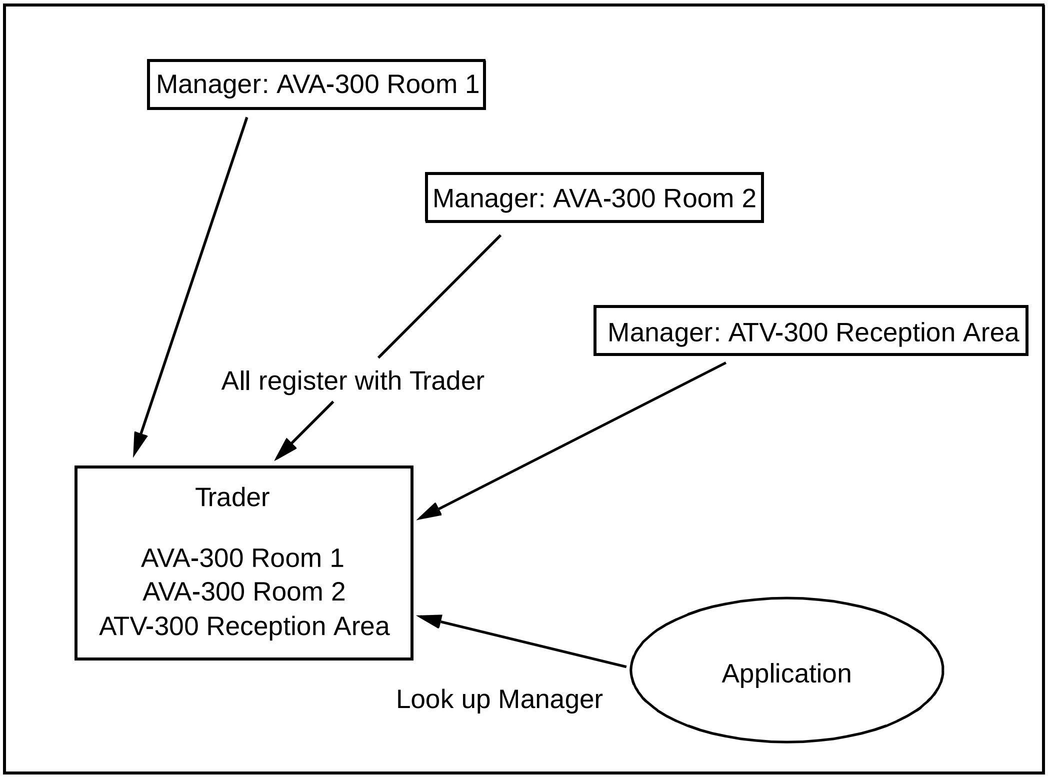

Figure 7.2 - How Managers, Traders, and Applications Interact

Managers register with one or more traders. Applications must look up managers in the trader to determine where the managers are located before they can invoke any operations on the manager's RPC interface.

7.3 Managers and the ATM Network

Managers implement ATM signalling on behalf of the devices they manage using a technique called proxy signalling. Proxy signalling relies on the devices being able to forward all signalling messages they receive from the ATM network to their manager, and on the manager being able to send signalling messages to the ATM network via the device. This means that the device appears to be the producer and consumer of all signalling messages sent to and received from the ATM network. The network is unable to distinguish whether the device itself is implementing ATM signalling, or if the manger is doing so on behalf of the device.

The communication paths required for forwarding signalling messages to and from the device are provided by Permanent Virtual Circuits (PVCs). Once configured, PVCs are remembered by the network and require no further interaction with the signalling protocols. The SVA software supports two types of ATM signalling:

- ATM Forum's UNI 3.0
- ATM Forum's UNI 3.1

The ATM Forum is an industry consortium created to accelerate the development and deployment of ATM standards; UNI 3.0 was the first version of their standard for ATM signalling.

NOTE UNI 3.0 and UNI 3.1 signalling may co-exist on the same network.

Both UNI signalling protocols provide similar functionality, so SVA managers can provide the same service regardless of which version is being used. SVA managers, which are applications from the network's point of view, implement both versions.

Deciding which protocol to use is based on the command line arguments given to the manager when run. It is not possible to dynamically switch between UNI 3.0 and UNI 3.1 at run time.

7.3.1 Switched Virtual Circuits (SVCs)

PVCs are used for signalling and private communication between a device and its manager. All other communication between devices and applications is typically achieved through dynamically-created Switched Virtual Circuits (SVCs); all video and audio data is sent over SVCs. An important function provided by managers is the dynamic creation of SVCs between their client applications and the device they each manage.

Managers make use of multicast SVCs to optimize the use of network resources. That is, if more than one application requests the same video or audio stream, the manager creates a single multicast SVC to carry the data to all interested applications. The use of multicast SVCs ensures that the same data is not sent multiple times over the same shared network link.

Consider a video stream being generated by an AVA-300 which is being received by three workstations. The network configuration is such that the workstations are connected to the same ATM switch, but there is a second switch between the AVA-300 and the workstation switch. Figure 7.3 illustrates this configuration.

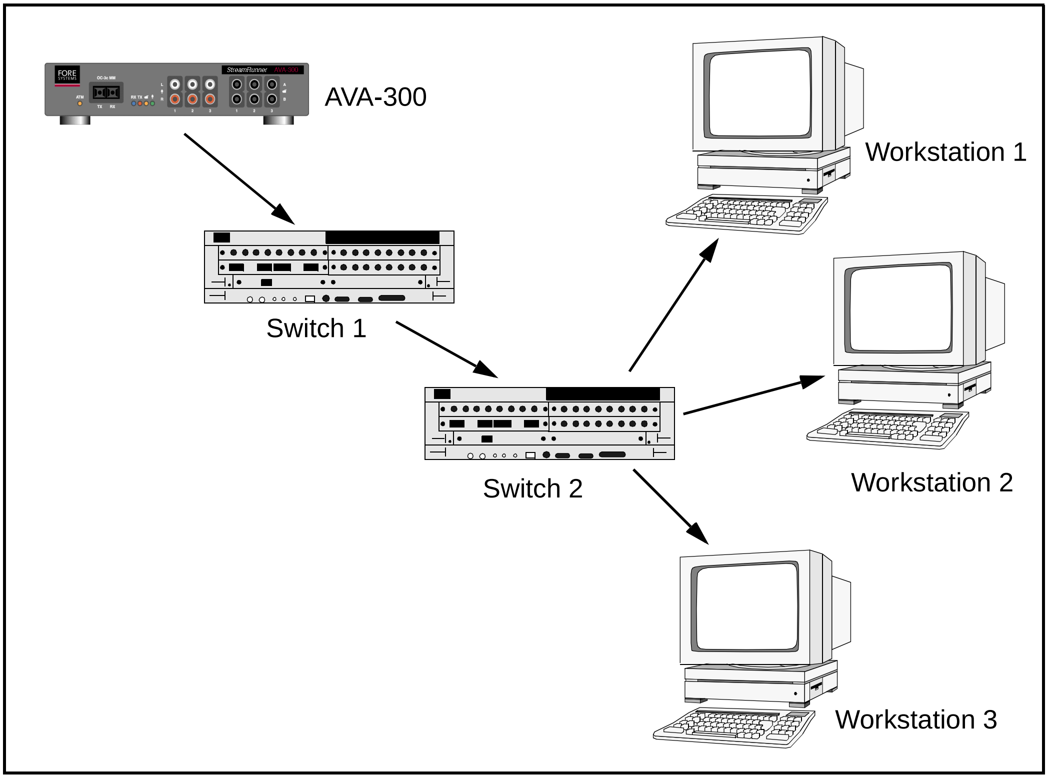

Figure 7.3 - Video and Audio Multicast

SVA Software

Multicasting means that only a single video stream is required between the AVA-300 and the switch to which it is directly connected. The switch that the workstations are connected to creates or copies a separate video stream for each workstation receiving the video. This reduces the load on the network (i.e., only a single video stream between switches) and also on the AVA-300 since it can generate a single, high-quality video stream which can be received by multiple workstations.

7.3.2 Permanent Virtual Circuits (PVCs)

Although SVCs are the preferred means of carrying video and audio data, there are certain circumstances in which their use is not possible. For example, you may encounter an ATM switch that does not support SVCs or you may wish to send video and audio streams to a non-SVA-compatible device or software application that does not fully support SVCs.

To address these situations, the SVA software allows the use of Permanent Virtual Circuits (PVCs) to carry video and audio streams from an AVA-300 to a client application, ATV-300, or any other device or application.

NOTE

A single AVA-300 may source SVC and PVC streams simultaneously.

7.3.3 Manager Downlink

In addition to the use of ONC RPC to communicate between managers and their clients, managers maintain a downlink to each of their clients. The downlink is used to inform clients of changes in the configuration of video or audio streams. This allows clients to dynamically adjust their presentation of these streams to match the current configuration. This is particularly important when one application is editing a stream which is being received by applications running on other workstations. The edits made by the editing application are immediately propagated to the other applications over the downlink.

The downlink is implemented using the Internet UDP/IP protocol. The manager sends downlink messages to each of its clients using UDP datagrams.

NOTE Since this scheme does not scale well to large networks, an alternative ATM multicast SVC downlink implementation is also supported.

The use of a multicast connection means that the manager can support a very large number of clients since it does not have to maintain a separate network connection to each one. In addition, traffic through the network is kept to a minimum.

An environment variable **`SVA_ATM_DOWLINK`** can be set to `On` to force the use of the ATM downlink. If the manager fails to establish a multicast SVC connection for the downlink to a particular client application, it falls back to using the Internet UDP/IP protocol instead. This means that it is possible to run client applications on workstations which are not connected to an ATM network; such applications cannot receive video or audio streams, but can manage AVA-300s and ATV-300s, and edit stream definitions.

Both native ATM and UDP downlinks can be serviced concurrently by a single manager. The UDP downlink is also used when SVCs are not supported, such as in a PVC-only ATM network.

7.4 CellChain™

The CellChain™ is a mechanism that allows a number of ATM devices to share the facilities of a single ATM switch port. Each device sharing the switch port is linked to the others by a unidirectional ATM transmission link. The high bandwidth available on a single ATM switch port and the low data rates that many ATM devices require make it attractive for a number of devices to share the same switch port.

An ATM switch port consists of a transmit and receive interface. Each ATM device that connects to the switch also has a transmit and receive interface. In a CellChain system, the transmit interface from the ATM switch port is connected to the receive interface of the head device in the CellChain. The transmit interface from the head device is then connected to the receive interface of the second device in the CellChain. This process continues until the transmit interface from the final device on the CellChain is connected back to the receive interface on the switch port to which the head device is connected.

Figure 7.4 illustrates an ATM device CellChain consisting of three AVA-300s. Cells coming from the ATM network for `Unit 2` must be forwarded on by `Unit 0` and `Unit 1`. Similarly, cells destined for the ATM network that originate on `Unit 0` must be forwarded by `Unit 1` and `Unit 2`. Devices are either positioned “upstream” or “downstream” on a CellChain. For example, in Figure 7.4, `Unit 0` is upstream of `Unit 1` and `Unit 2` is downstream of `Unit 1`.

NOTE Each device only has a single, standard ATM interface.

NOTE CellChain configuration is not currently supported on AVA-300s and ATV-300s that do not have fiber optic physical interfaces.

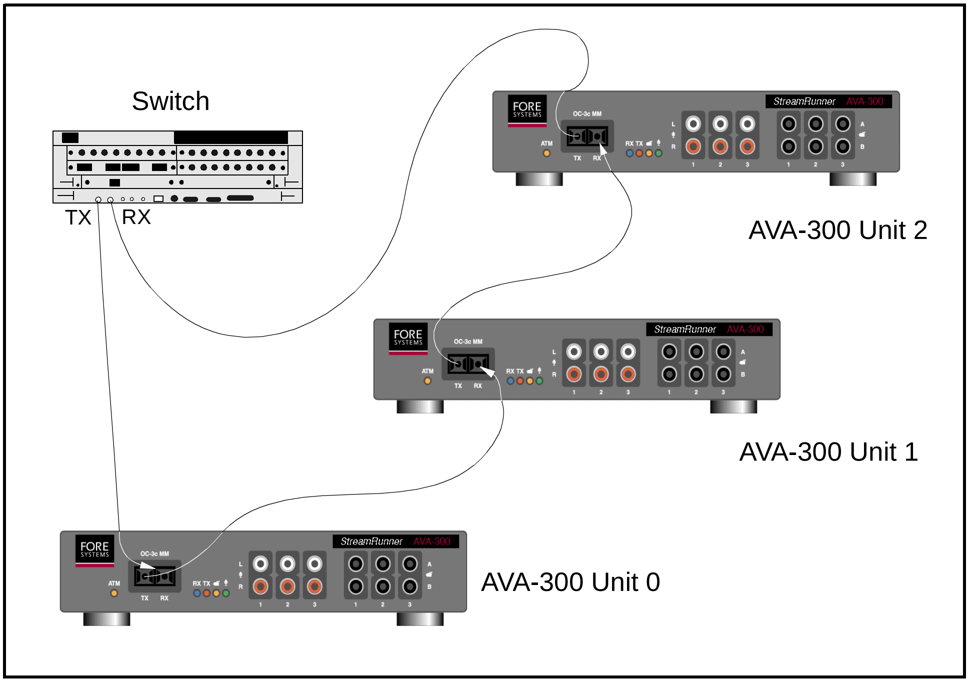

Figure 7.4 - ATM Device CellChain

Since the interconnect between the devices uses standard ATM technology, the inter-device distance is constrained only by the limitations of the particular ATM physical layer employed. The ATM Forum standard for framing ATM cells over SONET OC3c (155 Mbps) includes multimode or single mode fiber. In a case, for example, where multimode fiber is used to link the devices, each unit could be separated by up to 2000m with no intervening regenerator plant.

7.4.1 Topology Discovery

To control the device CellChain function, the `svamgr` process (described and outlined in Chapter 4) must know the type and position of all the devices on the CellChain. This is achieved by `svamgr` sending a topology discovery packet to the head device on the CellChain. The head device writes its own identifier into the packet before forwarding it to the succeeding device on the CellChain. When the packet finally exits the CellChain, all the devices will have written their identifiers into it.

7.4.2 Supported CellChain Configurations

The CellChain mechanism relies on the ability of the devices in the CellChain to forward cells which are not destined for themselves to the next device in the CellChain at high speed.

NOTE Only a single ATV-300 should be connected in a device CellChain and, if present, it must be the head device.

NOTE The SVA software will issue a warning if an invalid configuration is attempted.

NOTE The limit on the length of a CellChain configuration is 3 AVA-300s and 1 ATV-300.

NOTE If a CellChain is configured to send more than 90 Mbps of traffic, loss of video quality may be observed.

7.4.3 CellChain Addressing

The SVA software does not support the naming of individual units on a CellChain. Instead, the entire CellChain is assigned the name that is supplied to the `svamgr` process as a command line argument. An individual unit on the CellChain is addressed by unit number. The head device on the CellChain is designated as `Unit 0`. The second device on the CellChain is designated as `Unit 1`, and so on.

When processing its stream configuration startup file the `svamgr` process will, if no specific unit is specified for the stream, choose the first device on the CellChain which is capable of supporting the stream definition supplied.

It is possible to use `svc-rtds` to edit a stream definition so that the unit number is changed. The SVA software can move an active multicast call from one unit to another on the CellChain without breaking any of the connections.

7.4.4 Intra-CellChain Communication

It is not generally possible for devices on the same CellChain to exchange information. For example, it is not possible for an ATV-300 on a CellChain to receive and display a video stream from an AVA-300 on the same CellChain. This would require the switch port to loop back cells it receives directly back out onto the same port, rather than out to a different port on the same switch. This kind of functionality is unusual for ATM switches and therefore cannot be relied upon.

The ability of a downstream device to receive data from an upstream device on the same CellChain, while technically possible regardless of which switch implementation the CellChain is connected to, is not currently supported in the SVA software distribution.

7.4.5 Creating a CellChain Setup

To create a CellChain comprising three AVA-300s, do the following:

1. Connect one end of a fiber optic cable to the transmit port (`Tx`) on the front panel of the switch. Connect the other end of the fiber optic cable to the receive port (`Rx`) on the front panel of `Unit 0`.

NOTE To implement CellChain, you must use AVA-300 firmware version 4.3 or greater and ATV-300 firmware version 300.13 or greater.

NOTE This example assumes that port **`1A1`** on switch **`orion`** is being used to create the CellChain configuration.

2. Connect one end of a fiber optic cable to the transmit port (`Tx`) on the front panel of `Unit 0`. Connect the other end of the fiber optic cable to the receive port (`Rx`) on the front panel of `Unit 1`.
3. Connect one end of a fiber optic cable to the transmit port (`Tx`) on the front panel of `Unit 1`. Connect the other end of the fiber optic cable to the receive port (`Rx`) on the front panel of `Unit 2`.
4. Connect one end of a fiber optic cable to the transmit port (`Tx`) on the front panel of `Unit 2`. Connect the other end of the fiber optic cable to the receive port (`Rx`) on the front panel of the switch.
5. Verify the physical CellChain setup by typing:

   `svarun listchain -device orion:1A1`

   The system displays the following information:

   ```
   listchain: 3 devices (AVA AVA AVA)
   ```

6. Start a manager using the method that is appropriate to your signalling protocol. Refer to Section 4.6.2.

   At this point, `svamgr` issues a topology request to each device on the CellChain. When configured properly, `svamgr` displays a topology report similar to the `listchain` information:

   ```
   svamgr: 3 devices (AVA, AVA, AVA)
   ```

   You have created a CellChain comprising three AVA-300s.

7.5 Video and Audio Concepts

This section discusses the video and audio concepts essential to stream configuration and transmission. These concepts are used in practice with the SVA software distribution's Real-Time Display Software (`svc-rtds`) which is described in detail in Chapter 8.

7.5.1 ATM Adaption Layer 5

ATM Adaptation Layer 5 (AAL 5) is used for the video and audio stream and network control data sent and received by AVA-300s and ATV-300s. AAL5 provides efficient, variable-sized data transfer over ATM and imposes minimal overhead on the workstation applications which may process video and audio data.

AAL5 supports variable-sized packets or Protocol Data Units (PDUs). You must decide how much video or audio data to place in each PDU. A large PDU, while using network bandwidth more efficiently, takes longer to assemble and receive. This increased latency may be too large for real-time audio communication. Large PDUs also mean more stream data is lost if and when the PDU is lost. A small PDU uses more network bandwidth and increases the load on the receiver, but provides much lower latency. You must, therefore, strike a balance between large and small PDUs. A maximum PDU size of 4k bytes provides an acceptable trade-off between network utilization, data loss, and latency for all video formats and for all but the lowest data-rate audio formats. This upper limit is called the Maximum Transfer Unit (MTU).

7.5.2 Packing Factors

The amount of video or audio data that is placed in a single PDU is referred to as the Packing Factor and is expressed as the number of samples placed in each PDU. The size and format of each sample depends on the video or audio format being used. Thus, the packing factor depends on the format being used and the currently-specified MTU.

7.5.3 Peak Cell Rate

Video streams can generate very large amounts of data which can easily overload a single link in an ATM network or any receiving workstation. Even at lower frame rates, resolutions, and image quality, it is possible for the instantaneous data rate to be very high. The instantaneous data rate is the amount of data seen over a very short period of time (such as 1 millisecond). A data rate of 1Mbits/second, if transmitted in its entirety in 0.01 seconds, would give an instantaneous rate of 100Mbits/second.

All AVA-300 video stream formats require a Peak Cell Rate (PCR) definition, which limits the amount of ATM network bandwidth that the video stream consumes. You do not need to specify a PCR for audio streams since they are inherently constant bandwidth sources.

7.5.4 Video Standards

The two principal broadcast TV video standards supported by the AVA-300, ATV-300, and SVA software distribution are as follows:

PAL (50Hz) Used primarily in Europe. 768x576 resolution at 25 frames/second.

NTSC (60Hz) Used primarily in the U.S. and Japan. 640x480 resolution at 30 frames/second.

A frame consists of 576 or 480 numbered scan-lines of video pixels. Each video frame is transmitted as two fields each of half the vertical resolution of the frame. The fields are named “Odd Field” and “Even Field” since one consists of all the odd numbered scan-lines in the frame and the other of all the even numbered scan-lines. For 50Hz PAL, each field has a resolution of 768x288 and there are 50 fields/second; for 60Hz NTSC there are 60 640x240 fields/second. Each field is captured once every 1/50 or 1/60 of a second (for PAL and NTSC, respectively) and, therefore, each reflects the image at a different point in time as well as space. A TV receives each field in turn and displays them in their corresponding position on the screen. Because two fields are displayed simultaneously, this technique is called interlacing.

AVA-300s can be configured to transmit either interlaced or non-interlaced video streams. All video streams are transmitted onto the ATM network as a sequence of fields.

7.5.5 Video Formats

Regardless of the video format selected, each video field is broken down into tiles, each of which is 8x8 pixels, prior to transmission over the ATM network. Tiles are transmitted in left-to-right, top-to-bottom order; that is the top left-hand corner is transmitted first, the bottom right-hand last. The available video formats fall into two categories: uncompressed and compressed.

7.5.5.1 Uncompressed Video

The main uncompressed formats are follows:

8-bit Monochrome Each pixel is sent as 8 bits of luminance.

16-bit RGB Each pixel is sent as 5 bits of Red, 5 bits of Green, and 5 bits of Blue, with a pad bit to round up to 16.

24-bit RGB Each pixel is sent as 8 bits of Red, Green and Blue.

These formats offer a trade-off between quality and bandwidth. For example, 24-bit RGB gives the best quality, but requires the most bandwidth. 8-bit monochrome provides lower quality, but requires less bandwidth.

7.5.5.2 Compressed Video

Motion-JPEG (M-JPEG) compression is used to reduce the data rates required for high quality (close to 24-bit RGB) video. JPEG was defined by and takes its name from the Joint Photographic Experts Group which is a joint ISO/CCITT technical committee whose goal has been to develop a general-purpose standard for the compression of continuous tone (grayscale or true color) digital video images. Although originally intended for still images, it is easily extended to video. M-JPEG refers to the technique of using JPEG to compress each video field independently of any other (i.e., as a still image) and transmitting it in sequence.

M-JPEG gives variable amounts of compression depending on the image being compressed. A complex image (such as a tree) will not compress as well as a simpler image (like a single color studio background). Typical data rates fall in the range of 4-20Mbits/second for 25 frames (i.e., interlaced) video at PAL resolution (768x576). However, it is possible for an M-JPEG compressed image to burst at a much higher rate than this average (e.g. 70Mbit/s). If the compressed data rate exceeds the Peak Cell Rate (see Section 7.5.3), then the AVA-300 will automatically slow down the frame rate to avoid exceeding the currently set peak rate. This simple facility ensures that the AVA-300 will always meet Quality of Service (QoS) contracts set for it.

The quality of the compressed video (the amount of information lost by the compression process) can be controlled via the Q-Factor parameter. The lower the Q-Factor, the less information is lost and the better the video quality, which means a higher data rate. Q-Factor 20 gives near 24-bit RGB video quality. Q-Factor 32 gives slightly lower quality, but will be largely indistinguishable from Q-Factor 20 for most video scenes. Higher Q-Factors give progressively lower quality video.

Where information is lost, visual artifacts are introduced by the compression process. Such artifacts are generally most apparent around sharp edges such as caption lettering or straight lines (fence wiring, for example). The edges of the letters may appear blurred and the lines jagged. For most video clips, the artifacts will not be readily visible.

7.5.5.3 Variable Q-Factor

Later versions of the AVA-300 firmware (4.3 and above) support dynamic Q-Factor adaption. This facility allows the AVA-300 to adjust the Q-Factor dynamically on a frame-by-frame basis. When using Variable Q-Factor, in addition to the Peak Cell Rate control, the AVA-300 is also configured with a target compressed field size. If the AVA-300 detects that the current transmitted image sequence is requiring less data than the target field size, it will attempt to decrease the Q-Factor (thereby increasing the compressed field size and quality). Similarly, if the transmitted image sequence is requiring more data than the target field size, the adjustment will work in the opposite direction.

The benefit of Variable Q-Factor is that it allows the AVA-300 to be configured to send the best possible quality video image within the constraints of the QoS parameter associated with the ATM circuit used to transmit the encoded stream.

Consider the example of a remote lecture in which the speaker is using pre-recorded video clips at certain points in the presentation. It is likely that the pre-recorded material will generate larger compressed images than the simpler image of the lecturer. The ATM circuit used for the video stream may be configured with a QoS that is optimal for the lecturer. When the lecturer switches over to the pre-recorded material, the AVA-300 adjusts the Q-Factor accordingly so the highest quality image is produced while still maintaining the QoS objectives of the circuit.

NOTE

It is strongly recommended that Variable Q-Factor is used over Fixed Q-Factor on AVA-300s that support it.

NOTE

A single AVA-300 cannot simultaneously produce both Variable Q-Factor and Fixed Q-Factor streams.

7.5.6 Video Scaling

Video scaling allows you to specify different resolutions for the video sampling region and for the video display region. The AVA-300 supports video scaling of the input prior to transmission over the ATM network. The sampling region represents the source video image to be scaled and the display region represents the resulting scaled image. If the display region is smaller than the sampling region, the video is scaled down. Figure 7.5 illustrates the various scaling parameters and their use.

NOTE The AVA-300 can only scale down video. It cannot increase the size of a video region.

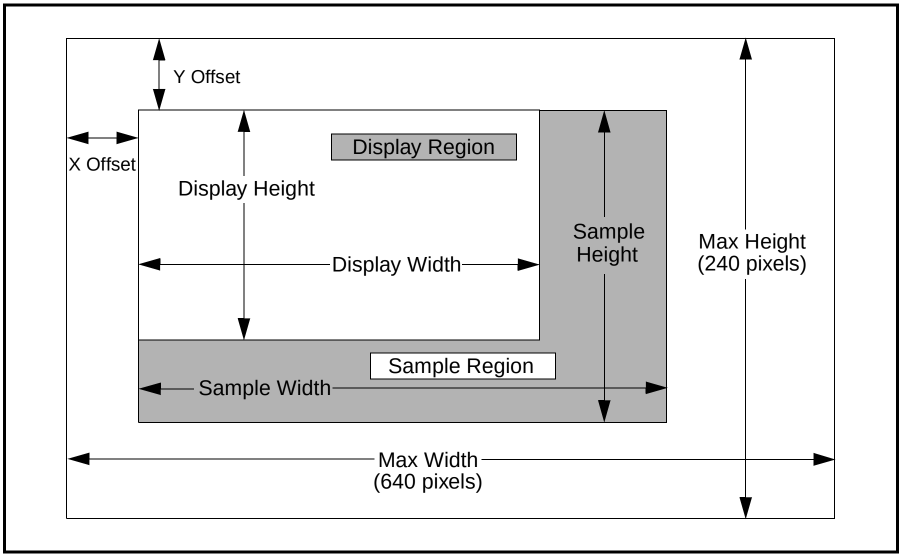

Figure 7.5 - Video Scaling Parameters (NTSC Example)

The parameters are as follows:

X & Y Offset Represents the position of the top left of the sample region in the original frame.

Sample Resolution Represents the size of the sample region.

Display Resolution Represents the size of the display region.

Maximum Resolution Represents either 768x288 (PAL) or 640x240 (NTSC).

7.5.7 AVA-300 Video Schedule

The AVA-300 allows the connection of multiple analog video inputs. The SVA software allows the creation of multiple video streams which are concurrently active from a single device. However, the AVA-300 only has a single video digitizer chip on-board, so it is not possible to encode multiple inputs each at full frame rate; rather the frame rate must be shared between all active inputs.

Each AVA-300 implements a schedule which comprises a sequence of slots, each of which indexes a video stream definition. A schedule is defined to last for one second; there is a slot for each video field in that time. An NTSC AVA-300 video schedule comprises 60 entries, as shown in Figure 7.6; a PAL schedule would comprise 50 entries.

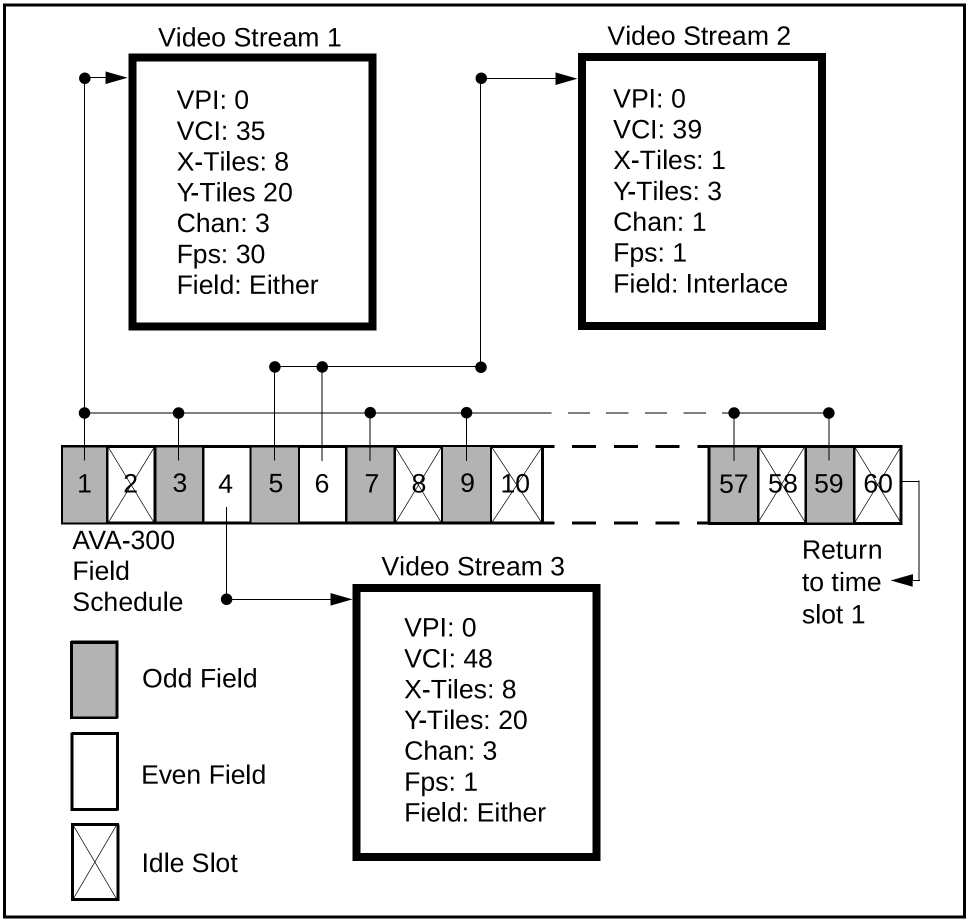

Figure 7.6 - NTSC AVA-300 Video Schedule

Each slot entry indexes a description of the characteristics of the video frame which is to be captured and sent over the network in the current schedule time. Different definitions may be referenced in successive field times. If an interlaced video stream is required for transmission, both the `Odd` and `Even` fields for a particular frame are digitized.

The video schedule is computed by the SVA software based on the currently-defined video streams. Once computed, the schedule is downloaded to the remote AVA-300. It is possible to request a video stream configuration for which it is impossible to calculate a schedule. For example, two interlaced video streams at full-frame cannot be supported. In this case, the SVA software tries to allocate the available resources fairly between the conflicting video stream definitions.

7.5.8 Video Input Synchronization

The AVA-300 allows the connection of up to 3 S-Video and 6 composite video inputs. If multiple video inputs are connected and they are “gen-locked,” they may be used simultaneously. If the video inputs are gen-locked, the AVA-300 may digitize one field from one input, the next field from another input, and so on. In this way, the AVA-300 may generate multiple video streams, each of which is coming from a different video input. A common configuration is to assign all odd fields to input 1 and all even fields to input 2 to give two full-rate, non-interlaced video streams.

NOTE

A gen-locked system is typically achieved by driving a set of video sources from a common sync signal. To accomplish this, each video source should have a “SYNC IN” input which is usually not standard on most devices. Alternatively, a “gen-lock” box, which takes multiple, non-synchronized video inputs and synchronizes them, can be used.

If the video inputs are not gen-locked, the AVA-300 can take up to eight fields to synchronize to a new input. This is similar to changing channels on a TV; the resulting video will appear to flash or be briefly skewed as the AVA-300 tries to lock onto the new input. This is not a problem if you are switching the input for a single stream; however, if you are trying to generate multiple video streams from several inputs which are not gen-locked, some or all of the resultant streams may lose synchronization.

NOTE

If your video devices are gen-locked, the AVA-300 can generate multiple video streams from any combination of the inputs.

If your video devices are not gen-locked, the AVA-300 can generate multiple video streams, but they should all use the same input.

7.5.9 Audio Formats and Parameters

The AVA-300 supports the following major audio formats:

A-Law	8-bit companded audio as developed for the US telephone system.
μ-Law	8-bit companded audio as developed for the United Kingdom and European telephone system.
16-bit linear PCM	16-bit linearly encoded audio used for CDs.

Each of these formats may be sampled at a variety of rates, the primary ones being:

8KHz	For standard telephone quality
44.1KHz	For CD quality
48KHz	For DAT quality

Both mono and stereo is supported for each format. However, the 8-bit modes sampled at 8KHz are generally only configured as mono for compatibility with older workstation audio playback hardware.

The data rates required for audio depend on the format chosen, the sampling rate, and whether the stream is configured as mono or stereo.

The low data rates for the 8-bit modes sampled at 8KHz have an important implication for packing factors (see Section 7.5.2). An MTU of 4K bytes (32Kbits) contains half a second's worth of 8-bit mono audio samples. This leads to unacceptable delay and noticeably poorer audio quality. Consequently, a smaller MTU (of 1Kbytes) is better suited to these modes.

7.6 Quality of Service (QoS)

When a connection is created across an ATM network, it is possible to specify a Quality of Service (QoS) that describes the requirements of the traffic being transmitted. Through a variety of traffic management techniques, it is possible for the ATM network to guarantee different levels of service depending upon the requested QoS, for different connections. There are three basic QoS classes that are supported by the SVA software:

Constant Bit Rate (CBR) A dedicated Peak Cell Rate (PCR) is specified when the connection is made. The system guarantees the bandwidth allocation for the duration of the connection. CBR is recommended for time-sensitive traffic such as audio.

Variable Bit Rate (VBR) A dedicated PCR, Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS) are specified when the connection is made. The system guarantees a sustained bandwidth and peak maximum for the duration of the connection. VBR is recommended for compressed video traffic.

Unspecified Bit Rate (UBR) The system provides no bandwidth guarantees; the connection must use the bandwidth available at that point in time. A “best effort” service class, UBR is recommended for traffic that is not time-sensitive, such as broadcast information like e-mail.

The simplest way of using QoS with AVA-300s and ATV-300s requires that the devices be used under control of a manager process. When a manager process receives a request to join to a stream, it uses the stream QoS policy and stream attributes to compute the detailed QoS specification to implement in the ATM connection establishment request. The QoS policy associated with a stream may be altered using the appropriate `svc-rtds` edit command. The default QoS policy is for a stream to be configured as UBR. The `svc-rtds` application is fully described in Chapter 8.

When using a UBR stream definition, the stream will not be refused admission into the ATM network because of a lack of bandwidth resources. It may, however, be refused because of lack of other resources, such as VPI/VCI space, on a congested link. The SVA software sets the advisory PCR01 field in the UBR contract to be the PCR of the associated stream at connection establishment time.

When using CBR and VBR stream definitions, it is possible for the connection to be refused because of a lack of bandwidth/buffering resources in the ATM network. This issue is of particular concern for SVA video streams, and is discussed further in Section 7.6.7.

7.6.1 CBR Specification

The CBR contract selected by the SVA software specifies PCR conformance testing of the CLP=0 and CLP=0+1 cell streams by the network. The network typically uses dual leaky bucket hardware in order to achieve this. By default, the SVA software enables cell tagging in the CBR contract. Cell tagging may be disabled on a per-device basis by setting the `SVA_TAG_ENABLE` environment variable to zero prior to starting the device manager process.

The network uses the first leaky bucket to assess conformance to PCR of the CLP=0 cells. It uses the second leaky bucket to assess the conformance to PCR of the aggregate of the CLP=0 and CLP=1 cells. If cell tagging is enabled, the cells which fail the PCR CLP=0 test are tagged as CLP=1 and passed on to the second leaky bucket to be tested for PCR CLP=0+1 conformance. Cells that fail the PCR test on CLP=0+1 are discarded. If cell tagging is not enabled, cells which fail the PCR CLP=0 test are discarded and cells which fail the PCR test on CLP=0+1 are discarded.

7.6.2 VBR Specification

The VBR contract selected by the SVA software specifies PCR conformance testing of the CLP=0+1 cell stream and MBS and SCR conformance of the CLP=0 cell stream by the network. The network typically uses dual leaky bucket hardware in order to achieve this. By default, the SVA software enables cell tagging in the VBR contract. Cell tagging may be disabled on a per-device basis by setting the `SVA_TAG_ENABLE` environment variable to zero prior to starting the device manager process.

The network uses the first leaky bucket to assess conformance to PCR of the aggregate of the CLP=0 and CLP=1 cells. Cells that fail this test are discarded. The network uses the second leaky bucket to assess the conformance to SCR and MBS of the CLP=0 cells. If cell tagging is enabled, the cells which fail this test are tagged as CLP=1. If cell tagging is disabled, the cells which fail this test are discarded.

7.6.3 Serial Stream QoS

The serial data streams generated by the AVA-300 do not consume a great deal of bandwidth. In addition, since the serial data cells are sent out at evenly-spaced intervals, they are able to easily conform to a CBR traffic definition. If you are required to transfer an AVA-300 serial stream across the ATM network with a high guarantee of success, then it is recommended that you configure the serial stream with CBR QoS.

7.6.4 Audio Stream QoS

The audio streams generated by the AVA-300 are not compressed; i.e., the amount of data generated is not dependent on the audio sample values. Since audio is continually sampled and AAL5 PDUs are not buffered prior to transmission, an AVA-300 audio stream consists of a regular sequence of evenly-spaced cells. If you are required to transfer an AVA-300 audio stream across the ATM network with a high guarantee of success, then it is recommended that you configure the audio stream with CBR QoS.

7.6.5 Uncompressed Video Stream QoS

Uncompressed AVA-300 video is typically used when sending to a workstation or PC device. In this instance, you would typically use non-interlaced video and limited frame rates. Each video field is sent at a constant rate that is bounded by the PCR associated with the SVA video stream definition. If you are required to transfer an AVA-300 uncompressed video stream across the ATM network with a high guarantee of success, then it is recommended that you configure the video stream with VBR QoS.

In some instances you may wish to use a CBR video stream QoS definition if you need to ensure that the guaranteed bandwidth is allocated for the stream in the face of VBR overbooking (see Section 7.6.7).

7.6.6 Compressed Video Stream QoS

There are two categories of compressed video streams: fixed Q-Factor and variable Q-Factor. Since it is not advisable to implement fixed Q-Factor streams, little effort has been made in developing QoS profiles for this category. If you desire to use QoS and a fixed Q-Factor video stream, then it is recommended that you configure the video stream with CBR QoS.

The variable Q-Factor mechanism is ideal for translating into a VBR QoS specification. The variable Q-Factor stream definition comprises a peak cell rate and a sustained data rate parameter. These translate readily on to the PCR and SCR definitions required for the VBR QoS contract. By default, the SCR value allocated by the SVA software is 10% greater than that specified in the stream definition to allow for overshoot as the AVA-300 adapts the Q-Factor in extreme cases. This value can be altered on a per-device basis by setting the `VARQ_SCR_SCALE` environment variable prior to starting the device manager process. The default value is `110`.

By default, the MBS value is set at 50% the size of a target video field or frame (depending on whether the video stream is non-interlaced or interlaced, respectively). This value may be altered on a per-device basis by setting the `VARQ_MBS_SCALE` environment variable prior to starting the device manager process. The default value is `50`.

SVA Software

7.6.7 VBR Overbooking

By default, a FORE Systems ATM network will not overbook (sometimes referred to as statistically multiplex) VBR circuits. The admission control algorithm calculates an equivalent capacity for the requested QoS which is heavily weighted towards the PCR value that is specified. If the requested equivalent capacity causes the allocated capacity at a point in the network to exceed the link capacity, then the connection will be refused.

Specifying VBR overbooking tells the network to do the admission test against some multiple of the link capacity. To configure VBR overbooking in a FORE Systems ATM network, please refer to the *ForeRunner ATM Switch Configuration Manual*.

7.7 SVA Security

All components of the SVA software distribution come with security features that prevent unwelcome users from accessing streams to which they do not have authorization. These security measures are based on password protection. The security system operates on a per-manager basis.

7.7.1 Setting Up Security Measures

To use the SVA software's security features, you must create a password file containing a list of manager names and their associated passwords.

7.7.1.1 Password File Format

Password files have a simple ASCII format. A line may contain a comment, a password, or be blank. Lines whose first non-blank character is a pound sign (#) are regarded as comments, and, like blank lines, are ignored. Password lines have the following format:

```
<name>:<passwd>[:<comment>]
```

where `name` is the name of the manager, `passwd` is the password assigned to it, and `comment` is an optional comment field that is ignored by the security process. The fields must be separated by colons, tab characters, or a combination of both. Spaces are allowed in manager names and in the middle (but not at the ends) of the password. Valid passwords must be between 6 and 16 characters in length.

7.7.1.2 Password File Location

The default location of the password file `SVAauth` is in your home directory. The SVA software uses the environment variable `$HOME` to determine the location of your home directory. SVA applications use the default password file name and location unless explicitly directed elsewhere by using the `-passwdfile` switch which is an argument to the relevant SVA applications.

7.7.1.3 Password File Semantics

The password file performs a dual role:

- It is used by you (or your system administrator) to set the passwords for managers as they are started.
- It is used by others whenever they need to supply a password to gain access to a manager.

Your system administrator is likely to have a single password file for all the managers in the local system. Access to this file should be carefully restricted. As a local user, you maintain a password file in your home directory which only you may access and which contains the passwords for the managers to which you have access.

When you run `svamgr`, it attempts to locate a password file. When it finds one, the program looks for a password line matching its name. If it finds a match, it enables password protection. If it does not find a match, or if it fails to locate a password file, protection is disabled.

When a client program attempts to perform a restricted operation, such as opening a session on a password-protected manager, a password is required.

For interactive programs such as `svc-rtds`, a dialog is displayed asking you to type the password. If your password file contains this password, the password appears automatically in the dialog and does not need to be typed in.

For non-interactive programs such as `svapatch`, you do not have the option to type the password; it is essential that you have a password file containing the required lines.

7.7.2 Controlling Access

For the purposes of SVA security, you may either join or edit a stream. These two methods are available to different categories of users:

- You join a stream when you receive it or patch its video and/or audio (such as with `svapatch` or `svc-rtds` local sinks).
- You edit a stream when you change some or all of its parameters (e.g., using the `svc-rtds` stream editor).

The following three rules describe the stream access control system:

- You may join or edit streams from a manager whose password protection has not been enabled.
- You may join or edit a stream from a manager whose password protection has been enabled, provided the correct password is supplied.
- Anyone may join a public stream at a manager whose password protection is enabled without having to supply a password, unless another user holds a session (see Section 7.7.3) at that manager and has disabled `Public Streams`.

NOTE Permission to join and edit a stream is denied if none of the above rules apply.

7.7.3 Sessions

SVA managers support the concept of a user "holding a session" on a device. Only one user may hold a session on a manager at a given time. The purpose of the session is to act as a lock to ensure that certain features of the manager are not invoked by users simultaneously.

In order to obtain a session, the client program (such as `svc-rtds`) requires the manager's password, if any. If a session is not being held by anyone, requests to obtain one succeed; if one is already being held, any requests fail. In the case of failure, it is up to the client program to decide what to do; they are not prevented from going ahead and editing stream definitions, but most clients will report an error to the user.

When establishing a session, the client may choose to disable public streams, (in `svc-rtds` this is presented as an option to the user). The effect of disabling public streams is to make all streams private for the duration of the session. This can be used, for example, to have a private conversation over a normally public stream and prevent anyone from editing the stream.

7.7.4 SVA Security Guidelines

SVA security measures rely entirely on the normal UNIX files system access control features and the protection provided is only as strong as the protection afforded your files by the file system. Here are some suggested guidelines to follow concerning SVA security:

- Always keep your password file unreadable to other users. The UNIX mode for such a file is `octal 0600 (-rw-------)`.
- Be aware that certain editors leave behind backup files whose names are similar to the file being edited. These may be publicly readable.
- You should also be aware of other circumstances which may cause the password file to become publicly readable. For example, text editor core files and SVA application core files may contain the password file.
- Changing the access of a stream from public to private, either by using the stream editor or by opening a session, does not cause clients that joined the stream prior to the access change to be disconnected. Only subsequent join requests are denied.
- Many of the security mechanisms require the clients to be compliant, such as by denying users the edit method when they are unable to obtain the session. You should be aware that it is possible to write a client application that ignores the session holder and edits a stream to be public.

7.8 svapatch

`svapatch` is a client of the SVA manager RPC interface. It is used to connect a source stream from an AVA-300 to an ATV-300. In order for an SVC to be established between the two devices, a sink stream definition must be created in the destination device manager by `svapatch`.

Figure 7.7 illustrates some of the main RPC control paths that are used when establishing a multimedia SVC stream between an AVA-300 and ATV-300 using `svapatch`.

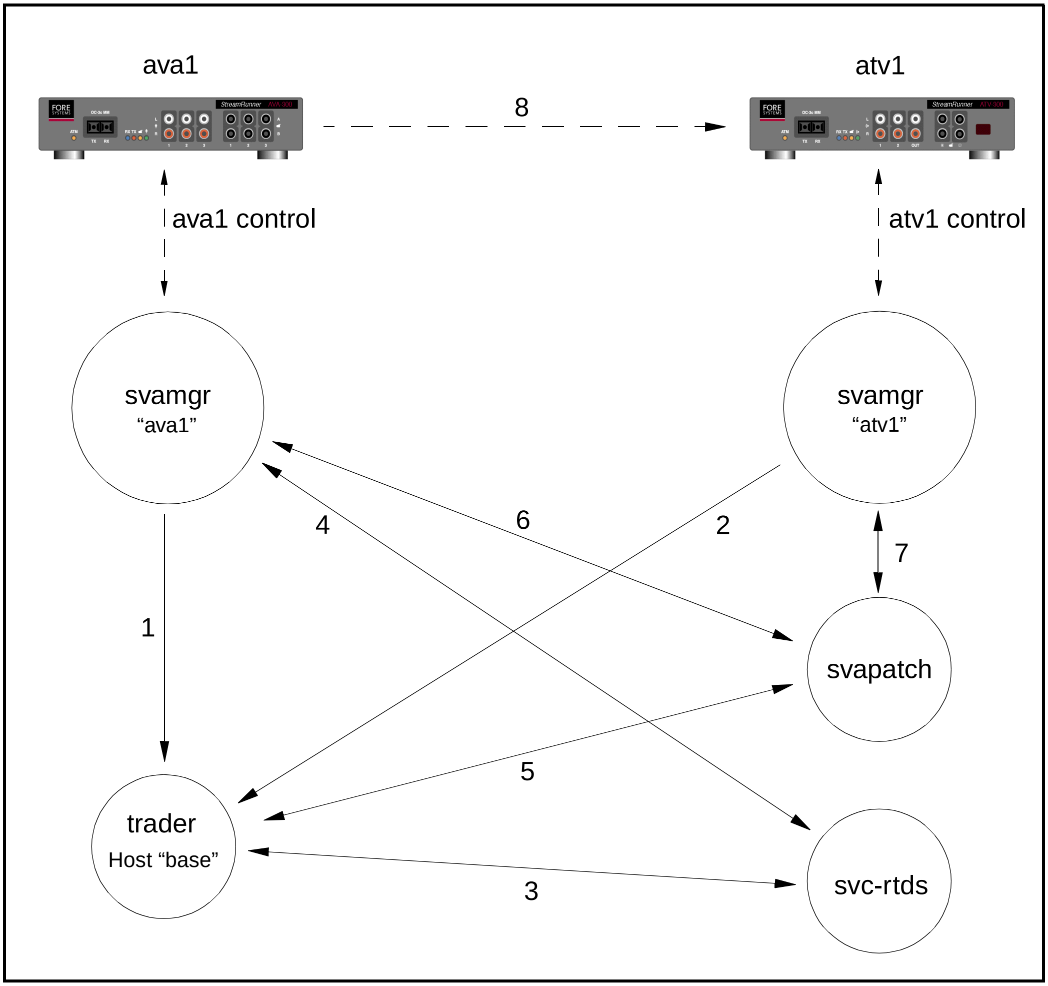

Figure 7.7 - RPC Paths and svapatch

Each circle represents an SVA process which may be running on any host in the network. Since all control communication is via RPC, the placement of these processes is decided by your local network administrator. A possible sequence of events, corresponding to the numbered arrows in Figure 7.7, for a connection to be established between the AVA-300 and ATV-300 is as follows:

1. The `ava1` manager process registers its interface with the `trader` process.
2. The `atv1` manager process registers its interface with the `trader` process.
3. The `svc-rtds` process looks up the interface of the `ava1` manager process. This step is only necessary if the client wishes to create a new stream definition or modify an existing one prior to performing the patch.
4. The `svc-rtds` process interacts directly with the `ava1` manager process. This may be because the client is creating or modifying the source stream definition or simply checking that it has the expected parameters.
5. The client knows the type and name of the stream(s) to be patched. For example, a single video stream called `jpeg-interlace` could be selected. In this case, the required command line to perform the patch is:

```
svarun svapatch -trader base -from ava1 -to atv1 jpeg-interlace
```

6. `svapatch` contacts the `ava1` manager process to query the details of the video stream `jpeg-interlace`.
7. `svapatch` opens a session with the `atv1` manager process. A sink stream definition compatible with that for video stream `jpeg-interlace` is created.
8. The video stream SVC from the AVA-300 to the ATV-300 is created.

`svapatch` continues to run after creating the initial sinks and SVC connections. Synchronization information and stream changes that are received on the downlink connection from the source device managers are passed to the appropriate sink device managers. So, if `svc-rtds` is used to change the size of a source video stream, `svapatch` passes the size change through to the destination.

`svapatch` attempts to maintain the specified connections in the face of device manager restarts, stream deletions and re-creations, connections failures (such as switch failures) and other error conditions.

NOTE

When `svapatch` opens a session with the ATV-300 manager, another client cannot open a session to the same manager. Therefore, it is not possible to use two instances of `svapatch` to patch multiple streams to the same ATV-300.

7.8.1 ATV-300 Menu Control

The ATV-300's on-screen menu system can be used to initiate both local and remote actions. The local functionality is described in Chapter 3. A variety of SVA applications are provided which interact remotely with an ATV-300 user via the menu system. These applications require the ATV-300 to be under the control of a management process. The two primary applications for remotely controlling the menu system are `atvpatch` and `atvmenu`. A description of `atvmenu` is provided in the next section. The `atvpatch` manual page is located in Appendix A.

7.9 atvmenu

`atvmenu` allows you to generate menu boxes on a remote ATV-300 and allows an action to be associated with each menu item. Since these actions may be shell commands such as `svapatch` (see Section 7.8), you can use the ATV-300 menu to join the streams generated by remote AVA-300s. `atvmenu` also supports the creation of submenus through which you may navigate. The submenus can themselves be generated by arbitrary shell commands, allowing sophisticated control systems to be built.

This section presents three examples which illustrate how to use `atvmenu`:

- Answering a "Yes or No" question
- Combining `atvmenu` and `svapatch`
- Creating submenus and gaining independent control over multiple video windows

7.9.1 Using atvmenu to Answer a “Yes or No” Question

To create a menu to receive the answer to a "Yes or No" question remotely, type the following:

```
$ atvmenu -at tv@base -title ‘Accept?‘\
-item ‘Yes‘ -exit "echo yes" -item ‘No‘ -exit "echo no"
```


When you issue this command, `atvmenu` contacts the manager process for the ATV-300 named `tv` which has exported registration information to the trader running on the machine called `base`. `atvmenu` opens a session with the manager and downloads a menu with the title `Accept` and the choices `Yes` and `No` to the ATV-300.

Once the menu has been displayed onscreen, one of the two choices may be selected using the ATV-300 Remote. Pressing `OK` or `<Enter>` chooses the item next to the menu cursor. Alternatively, `digit` button `1` selects `Yes` and `2` selects `No`. Both menu items in this example specify the `-exit` key. This instructs `atvmenu` to leave the current submenu; since it is the top-level menu in this case, `atvmenu` itself exits after executing the shell command associated with the selected item. The shell commands in this example simply echo the remote user‘s choice to the standard output.

NOTE

When `atvmenu` opens a session with the ATV-300 manager, another client cannot open a session to the same manager. Therefore, it is not possible to use two instances of `atvmenu` to create multiple menu boxes for the same ATV-300.

7.9.2 Combining atvmenu and svapatch

To create a menu that combines the use of `atvmenu` and `svapatch`, type the following:

```
$ atvmenu -at tv -title ‘Channel Menu‘ \
-item "Camera" -main "svapatch -from camera jpeg-interlace" \
-item "VCR" -main "svapatch -from vcr jpeg-interlace"
```

This command creates an onscreen menu at the ATV-300 named `tv` with two items called `Camera` and VCR. The shell commands for each of these items run `svapatch`. When `Camera` is selected, a patch is created from the stream `jpeg-interlace` at the AVA-300 called `camera` to the ATV-300 called `tv`.

NOTE The `svapatch` command does not need to explicitly specify `-to tv`; when `atvmenu` runs a shell command, it exports the name of the ATV-300 in the environment variable SVA_AT_DEV, and `svapatch` uses the value as the default `-to` device. See the `atvmenu` manual page in Appendix A for full details of the environment variables that `atvmenu` exports.

NOTE `atvmenu` can share access with its session with the `svapatch` process it creates, but not with independently-created `svapatch` processes.

VCR creates a similar patch from the AVA-300 named `vcr`. If the remote user selects `Camera` and then selects VCR, it is like changing channels on an ordinary TV; the patch from the `camera` AVA-300 is removed before creating the patch from the `vcr` AVA-300. To make sure that this happens, the shell commands for the two items have both been tagged with the same `-main` key.

NOTE Since neither of the choices in this example is tagged with the special `-exit` key, `atvmenu` remains running as the remote user makes menu selections instead of exiting after the first choice.

7.9.3 Using atvmenu to create submenus and gain independent control over multiple video windows.

To create submenus and gain independent control over multiple video windows, type the following, assuming the file `inset.menu` exists (see below):

```
$ atvmenu -at tv -title ‘Main Window‘ \
-item "Camera" -main "svapatch -from camera jpeg-interlace" \
-item "VCR" -main "svapatch -from vcr jpeg-interlace" \
-item "Inset window menu..." -menu "cat inset.menu"
```

The first two items use `atvmenu` and `svapatch` as described in the previous section. The third item is tagged with the special key `-menu`. When selected, this key runs the shell command `"cat inset.menu"` and interprets the command‘s output as a new submenu to download to the ATV-300. Suppose that the file `"inset.menu"` has the following contents:

```
$ cat inset.menu
```

```
-title 'Inset Window'
-item "Camera" -inset "svapatch -from camera -format pip video-jpeg-pip"
-item "VCR" -inset "svapatch -from vcr -format pip video-jpeg-pip"
-item "Back" -exit
```

Selecting the `"Inset window menu..."` item on the top level menu displays a three-item submenu. The last item, `"Back"`, has the special `-exit` tag; choosing it takes the remote user back to the original `Main Window` menu. Pressing the < button on the Remote also returns your user to the `Main Window` menu. The first two items on the `Inset Window` submenu run `svapatch` to make patches from the picture-in-picture streams at the two AVA-300s named `camera` and `vcr`. These two items have the `-inset` tag. Whenever one of these items is selected, `atvmenu` ensures that the `svapatch` run by the other item is terminated first. Because the key used is distinct from the one used for the `Main Window` in the top-level menu, switching between patches for the inset window has no effect on the patches for the `Main Window`, and vice versa. There is nothing inherently special about the `-main` and `-inset` tags other than they are distinct. `atvmenu` ensures that there is only one command running at a time for each distinct tag. In this example, the contents of the submenu are fixed. Because submenus are created from the outputs of shell commands, more dynamic menus are possible. The `atvmenudemo` program in the SVA software illustrates this by producing menus similar to those in this example for any ATV-300 and pair of AVA-300s specified on its command line. The shell mechanism means that menus can also be altered at runtime in response to user actions.

CHAPTER 8 svc-rtds

This chapter explains the features and functions of `svc-rtds`, the SVA software distribution's Real-Time Display Software program. `svc-rtds` is a GUI-based program that allows you to create, display, and edit video and audio streams on your workstation. This chapter assumes that you have completed the tasks outlined in Chapter 4 or Chapter 5 and resumes the explanation of `svc-rtds` where the Basic Setup procedures left off.

8.1 AVA/ATV Manager Browser

Once your trader and manager are running, you can access the `svc-rtds` application by typing:

```
svc-rtds -edit
```

NOTE

You must include the `-edit` argument to be able to create and edit video and audio streams; without specifying `-edit`, you would only be able to view or listen to preconfigured video and/or audio streams.

The `AVA/ATV Manager Browser`, shown in Figure 8.1, is displayed.

Figure 8.1 - AVA / ATV Manager Browser Window

svc-rtds

The AVA / ATV Manager Browser contains a list of managers it receives by querying the specified trader or traders. Any time a manager is started, it will export its registration information to a set of specified traders.

NOTE

In the event a manager is assigned a name which is already being used, the `AVA/ATV Manager Browser` appends the network address and port number of the second manager to its name to distinguish between the two entries:

```
ava1
ava1 (117.87.31.4:09732)
```

The `AVA/ATV Manager Browser` performs this operation if the two managers are running concurrently or if the same manager is restarted before its previous entry times-out.

In addition to listing the available managers, the `AVA/ATV Manager Browser` includes the following buttons:

Access Manager Allows you to access the `AVA/ATV Manager Window` which displays a list of all video and audio streams configured in the specified manager.

Describe Allows you to access the `Manager Description Window` which provides a full description of the selected manager.

Traders Allows you to access the `Traders Window` which displays a list of all the traders that `svc-rtds` is configured to query.

Exit Allows you to exit the `svc-rtds` application.

Each window and its functionality is described in this chapter.

8.2 Accessing the AVA/ATV Manager Window

Pressing the `Access Manager` button or double-clicking on the manager name in the `AVA/ATV Manager Browser` allows you to access the `AVA/ATV Manager Window`, shown in Figure 8.2:

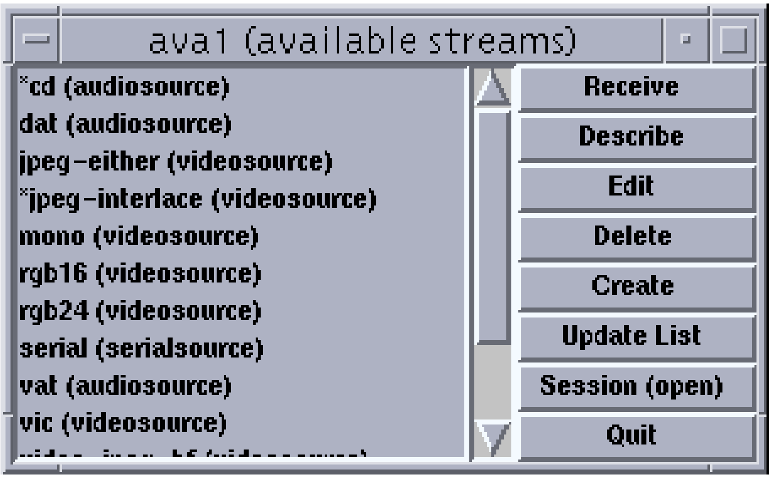

Figure 8.2 - AVA / ATV Manager Window

This window displays the audio and video streams configured in designated manager `ava1`. In addition, it displays the following buttons:

Receive Activates the highlighted video or audio stream and displays the `Video` or `Audio Stream Window` which allow you to view or listen to the selected video or audio stream.

Describe Displays either a `Video` or `Audio Description Window` which allows you to obtain a complete description of the selected stream.

Edit Displays either a `Video` or `Audio Edit Window` which allows you to edit the selected stream.

Delete Allows you to delete a highlighted stream.

Create Allows you to create a new video or audio stream.

Update List Allows you to update the AVA/ATV Manager Window with any additional stream information, much like a typical Refresh command.

Session (open/closed) Displays a Session Window which allows you to open and close sessions.

Quit Allows you to exit the AVA/ATV Manager Window and any associated description or edit window.

8.2.1 Accessing a Video Stream Window

To access a `Video Stream Window`, highlight the video stream you wish to access and click on `Receive`, or double-click on the stream name in the `AVA/ATV Manager Window`. The video stream is displayed within the `Video Stream Window`, as shown in Figure 8.3.

Click on the window to access the control buttons which are displayed at the bottom of Figure 8.3.

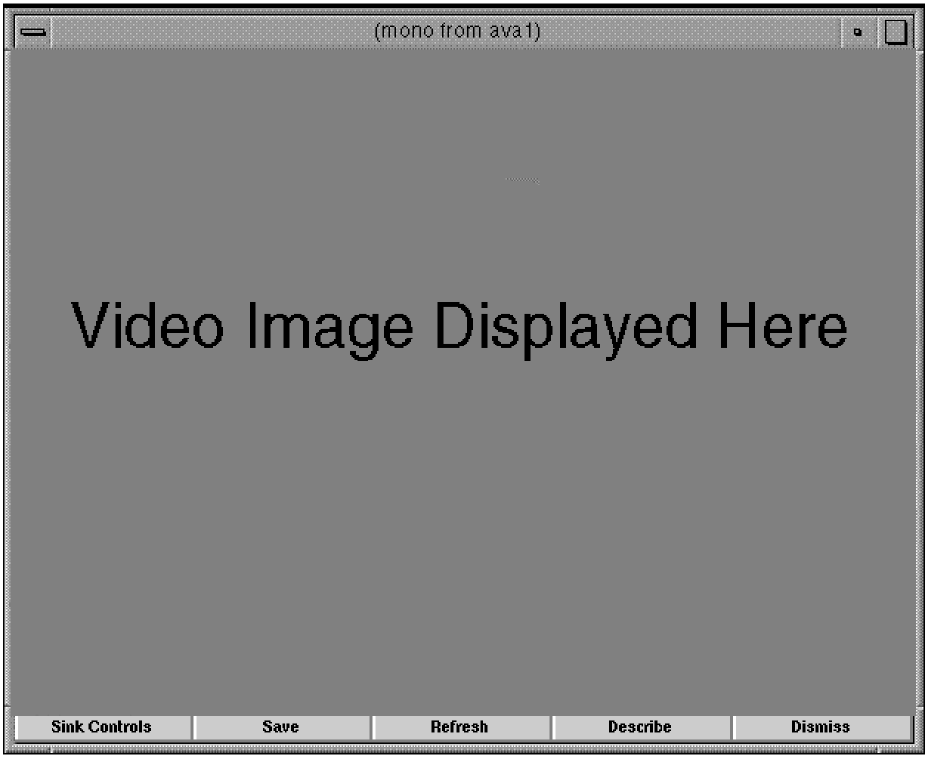

Figure 8.3 - Video Stream Window

8.2.1.1 Modifying Video Sink Controls

You can modify certain local aspects of the video sink from the `Video Stream Window` by clicking on `Sink Controls`. The `Video Sink Control Window`, shown in Figure 8.4, is displayed.

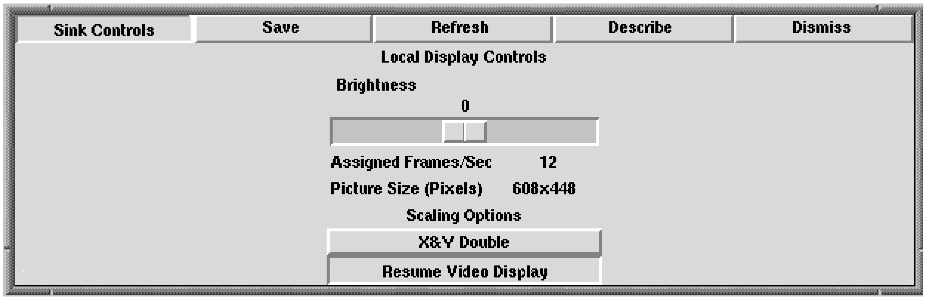

Figure 8.4 - Video Sink Control Window

From the `Video Sink Control Window`, you can modify the video stream's brightness and scaling options as follows:

Brightness Allows you to vary the relative intensity of the video image between very dim and very bright.

Scaling Options Allows you to adjust the video scaling vertically and horizontally. The options are `Small`, `Y Double`, and `X&Y Double`.

Pause/Resume Video Display Allows you to pause, and then resume, the video display.

NOTE Local sink control modifications cannot be saved; they only apply to a current viewing session and are lost when you exit.

NOTE Not all sinks support the `Brightness` control.

8.2.1.2 Saving Video Data

To save a single video frame or a sequence of frames to disk, click `Save`. The `Video Save Control Window`, shown in Figure 8.5, is displayed.

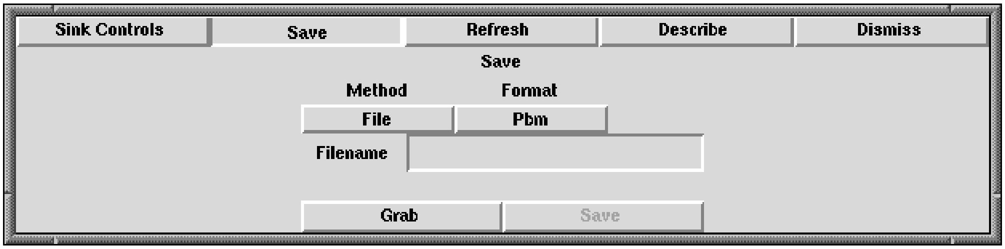

Figure 8.5 - Video Save Control Window

The save buttons on the `Video Save Control Window` have the following functions:

Method Allows you to access a pull-down menu that displays a set of methods you can use to save the video stream data. `File` lets you save the stream data to the file you name in the `Filename` field on disk.

Format Allows you to save video stream updates in either `JFIF` (for JPEG streams), `PBM` (for uncompressed streams), or `RAW` (for either format).

Grab Allows you to capture a single video frame.

Save Allows you to save a previously-grabbed video frame.

NOTE The `Grab` button does not operate if the video is paused (see previous page).

NOTE For JPEG streams, an extra `Record` button is provided which allows you to save each video frame to a separate file, prefixed by the specified filename.

8.2.1.3 Refreshing a Video Stream's Configuration Information

To refresh a video stream's configuration information, highlight the appropriate stream and click on `Refresh` in the `AVA/ATV Manager Browser,` or click on `Refresh` in any Video Control Window.

NOTE

Since managers automatically propagate configuration changes to `svc-rtds`, you will rarely need to perform a refresh. The refresh feature is provided in the event that automatic propagation is slow due to network congestion.

8.2.1.4 Exiting a Video Stream Window

To exit a Video Stream Window, click on `Dismiss`.

8.2.2 Accessing an Audio Stream Window

To access an `Audio Stream Window`, highlight the audio stream you wish to access and click on `Receive` in the `AVA/ATV Manager Window`. The audio stream is broadcast and an `Audio Sink Control Window`, as shown in Figure 8.6, is displayed.

8.2.2.1 Modifying Audio Sink Controls

You can modify certain local aspects of the audio sink from the `Audio Sink Control Window` which is displayed when you click on `Receive` in the `AVA/ATV Manager Window`.

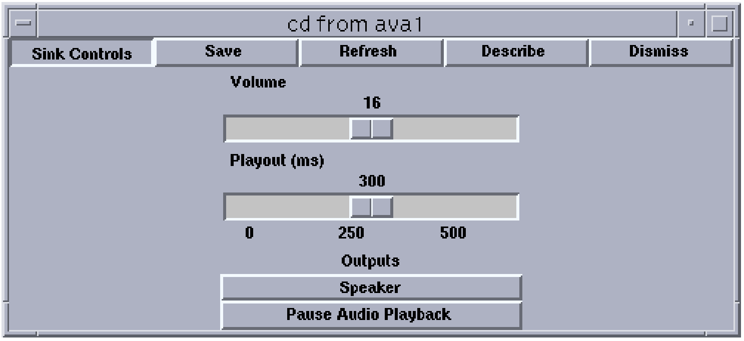

Figure 8.6 - Audio Sink Control Window

From the `Audio Sink Control Window`, you can modify the audio stream's volume, play-out, and output options as follows:

Volume — Allows you to vary the volume of the audio stream between very low and very high.

Playout — Allows you to adjust the audio stream's buffer latency.

Outputs — Allows you to select the audio stream's output source. The options are `Speaker`, `Headphones`, and `Line Out`, depending on the workstation/PC audio output capability.

Pause/Resume Audio Playback — Allows you to pause, and then resume, the audio playback.

NOTE

Local sink control modifications cannot be saved; they only apply to a current listening session and are lost when you exit.

8.2.2.2 Saving Audio Data

To save audio to disk, click `Save`. The `Audio Save Control Window`, shown in Figure 8.7, is displayed.

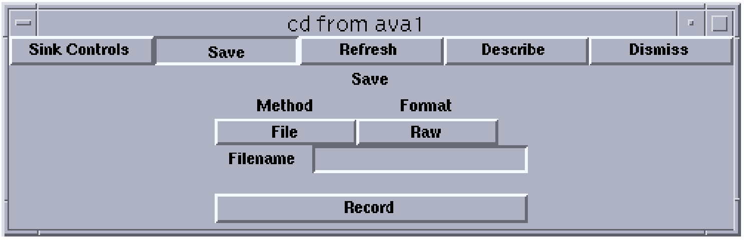

Figure 8.7 - Audio Save Control Window

The save buttons on the `Audio Save Control Window` have the following functions:

Method Allows you to access a pull-down menu that displays a set of methods you can use to save the audio stream data. `File` lets you save the stream data to the file you name in the `Filename` field on disk.

Format Allows you to save audio stream output to `RAW`, `WAV`, `AIFF`, or `SND` file formats.

Record Allows you to record audio output.

8.2.2.3 Refreshing an Audio Stream's Configuration Information

To refresh an audio stream's configuration information, highlight the appropriate stream and click on `Refresh` in the `AVA/ATV Manager Browser` or click on `Refresh` in any Audio Control Window.

NOTE

Since managers automatically propagate configuration changes to `svc-rtds`, you will rarely need to perform a refresh. The refresh feature is provided in the event that automatic propagation is slow due to network congestion.

8.2.2.4 Exiting an Audio Stream Window

To exit an Audio Stream Window, click on `Dismiss`.

8.2.3 Accessing a Video Description Window

The `Video Description Window`, shown in Figure 8.8, displays current information about a selected video stream.

To access a `Video Description Window`, highlight the video stream you require information about and click on `Describe` in the `AVA/ATV Manager Window`. The `Video Description Window` is displayed.

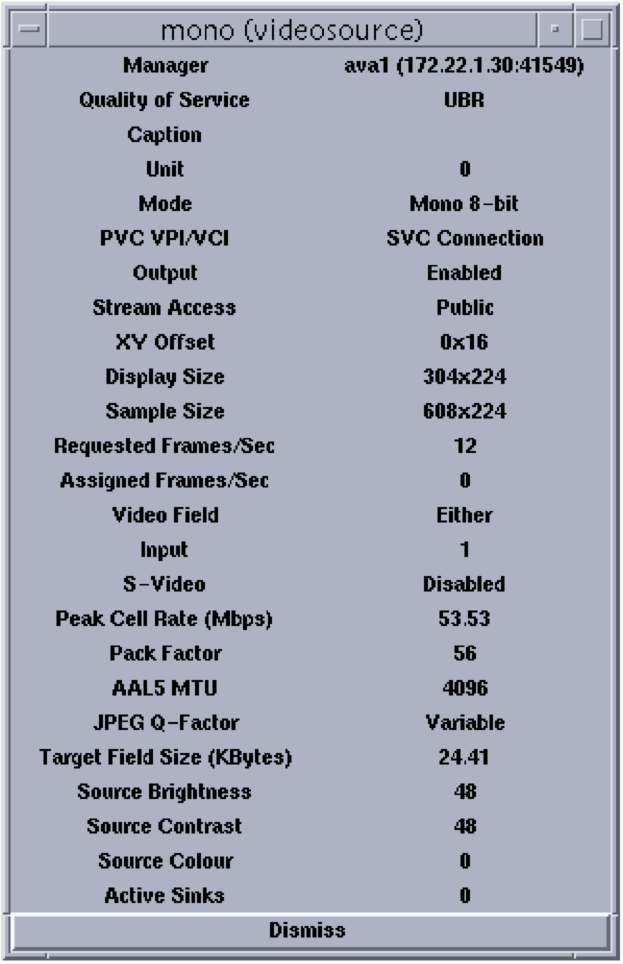

Figure 8.8 - Video Description Window

8.2.4 Accessing an Audio Description Window

The `Audio Description Window`, shown in Figure 8.9, displays current information about a selected audio stream.

To access an `Audio Description Window`, highlight the audio stream you require information about and click on `Describe` in the `AVA/ATV Manager Window`. The `Audio Description Window` is displayed.

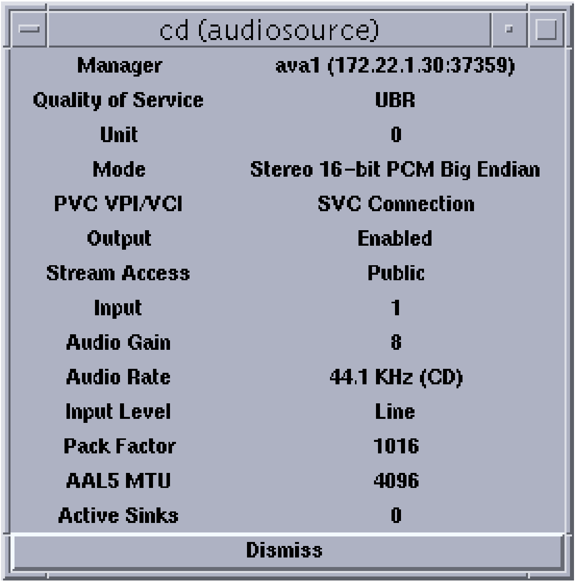

Figure 8.9 - Audio Description Window

8.2.5 Opening a Session

To edit, delete, or create video and audio streams, you must first open a session with the manager. You open a session by clicking on `Session` in the `AVA/ATV Manager Window`. The method you use depends whether or not access to the manager is password-protected.

8.2.5.1 Opening a Session when Manager is not Password-Protected

To open a session when the manager is not password-protected, do the following:

1. Click `Session` on the `AVA/ATV Manager Window`. The `Session Window`, shown in Figure 8.10, is displayed.

NOTE You are not required to enter a password when the manager is not password-protected, as indicated by the message "No password is set for this manager."

Figure 8.10 - Session Window, Not Password-Protected

2. Select the session attributes that you desire.
3. Click `Open` to open the session.

8.2.5.2 Opening a Session when Manager is Password-Protected

To open a session when the manager is password-protected, do the following:

1. Click `Session` on the `AVA/ATV Manager Window`. The `Session Window`, shown in Figure 8.11, is displayed.

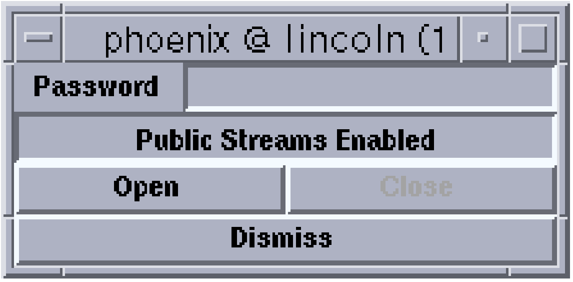

Figure 8.11 - Session Window, Password-Protected

2. Select the session attributes that you desire.
3. Type in your password.

NOTE If the `svc-rtds` password file already contains your password, it will automatically appear in the dialog box.

4. Click `Open` to open the session.

8.2.6 Editing a Video Stream

To edit a video stream, click on `Edit` in the `AVA/ATV Manager Window`. The `Video Edit Window`, shown in Figure 8.12, is displayed.

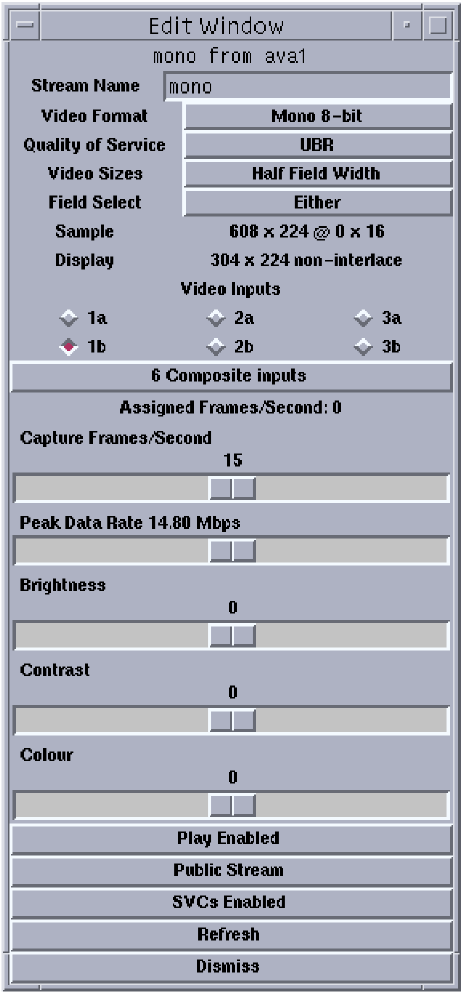

Figure 8.12 - Video Edit Window

NOTE

Before editing a video stream, you must first open a session.

The `Video Edit Window` has a number of fields you can edit to achieve the video stream quality you require. These include:

Stream Name Allows you to change the name of the video stream you have currently accessed.

Video Format Allows you to select from three compressed and uncompressed formats which include:
`Mono 8-bit`
`JPEG`
`RGB 15-bit Big Endian`
`RGB 15-bit Little Endian`
`RGB 24-bit 3-bytes`

Quality of Service Allows you to select the service class best suited to the traffic type. The available settings are:
`ubr` (default)
`cbr` (audio)
`vbr` (video) recommended

Video Sizes Allows you to scale the video image to fit your requirements. The available video sizes include:
`Eighth Field`
`Quarter Field Width`
`Quarter Field`
`Half Field Width`
`Full Field`
`Full Field (not clipped)`
`Other`

Field Select Allows you to select the video field sampling rate. The available settings are:
`Interlace`
`Either`
`Odd`
`Even`

Sample Displays the image's sample region dimensions.

Display Displays the image's display region dimensions.

Video Inputs Allows you to select from 6 composite input channels or 3 S-Video input channels. The input channel you select here must match the input used on the device's front panel.

Capture Frames/Second	Allows you to set your video stream rate depending upon your video format.
Peak Data Rate	Allows you to set the instantaneous and average data rates.
Brightness	Allows you to vary the relative intensity of the video image between very dim and very bright.
Contrast	Allows you to vary the range between the video image's white and black outputs.
Color	Allows you to vary the relative intensity of the red, green, and blue outputs.
Play Enabled/Disabled	Allows you to start or stop the video output at the source AVA-300.
Public/Private Stream	Allows you to designate a video stream as accessible (`Public`) or restricted (`Private`).
SVCs/PVCs Enabled	Allows you to specify either an SVC or PVC stream.
Refresh	Allows you to maintain settings that are consistent with the settings at the manager.
Dismiss	Allows you to exit the `Video Edit Window`.

M-JPEG streams also include these edit fields:

Q-Factor	Allows you to control compression and data rate.
Sustained Data Rate	Allows you to set the target data rate generated by the video stream. Appears when Variable Q-Factor is selected.

NOTE The Sustained Data Rate (SDR) value depends on the frame rate and whether or not you choose interlacing. For example, if you reduce the frame rate by half, the SDR is also reduced by half. The SDR display value is updated to reflect the modification.

NOTE For more information on stream configuration, refer to Section 7.5.

svc-rtds

8.2.7 Editing an Audio Stream

To edit an audio stream, click on `Edit` in the `AVA/ATV Manager Window`. The `Audio Edit Window`, shown in Figure 8.13, is displayed.

Figure 8.13 - Audio Edit Window

NOTE

Before editing an audio stream, you must first open a session.

The `Audio Edit Window` has a number of fields you can edit to achieve the audio stream quality you require. These include:

Stream Name — Allows you to change the name of the audio stream you have currently accessed.

Quality of Service — Allows you to select the service class best suited to the traffic type. The available settings are:
`ubr` (default)
`cbr`
`vbr`

Audio Formats — Allows you to select from the following formats:
`DAT`
`CD`
`A-Law`
`µ-Law`

Gain — Allows you adjust the device's volume level.

Input Select — Allows you to select from three stereo audio input channels. The input channel you select here must match the input used on the device's front panel.

Line/Mic Level — Allows you to set the audio input device type as either `Mic` for microphones that power themselves, or `Line` for CD player, VCR, etc.

Microphones that are designed to draw power from the equipment to which they are connected cannot be used. In addition, many self-powered microphones produce an output that is too low for use with the AVA-300. It is recommended that you use a microphone preamplifier that is designed for use with your microphone. The output from these preamplifiers is typically at Line level; experiment with both the `Line` and `Mic` settings on the AVA-300 for the best results.

Play Enabled/Disabled — Allows you to start or stop the audio output at the source AVA-300.

Public/Private Stream — Allows you to designate an audio stream as accessible (`Public`) or restricted (`Private`).

SVCs/PVCs Enabled — Allows you to specify either an SVC or PVC stream.

Refresh — Allows you to maintain settings that are consistent with the settings at the manager.

Dismiss — Allows you to exit the `Audio Edit Window`.

8.2.8 Deleting a Video or Audio Stream

To delete a video or audio stream, highlight the stream and then click on `Delete` in the `AVA/ATV Manager Window`. The `Delete Stream Window`, shown in Figure 8.14, is displayed.

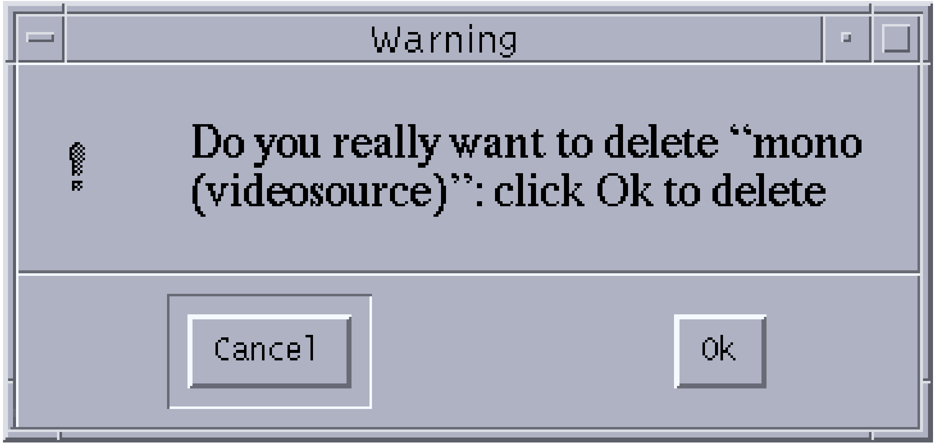

Figure 8.14 - Delete Stream Window

Click `Ok` to delete the stream from the manager or click `Cancel` to abort the process.

NOTE Before deleting a video or audio stream, you must first open a session.

8.2.9 Creating a Video or Audio Stream

To create a new video or audio stream, click on `Create` in the `AVA/ATV Manager Window`. The `Create Stream Window`, shown in Figure 8.15, is displayed.

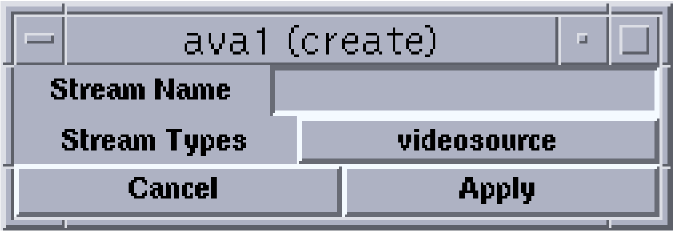

Figure 8.15 - Create Stream Window

1. Name the stream by typing in a meaningful name in the `Stream Name` field.
2. Designate the stream as either a video or an audio stream by selecting the appropriate designation in the `Stream Types` field.
3. Click on `Apply` to add the stream to the manager or `Cancel` to abort the stream create and exit the `Create Stream Window`.

You can apply the appropriate attributes to your newly-created stream by accessing either the `Video Edit Window` or `Audio Edit Window`.

NOTE

If an already-designated stream is selected in the `AVA/ATV Manager Window` when the create operation is invoked, the newly-created stream's initial attributes are the same as those of the selected stream.

8.2.10 Updating the AVA/ATV Manager Window

You can update the AVA/ATV Manager Window with new stream information by clicking on Update List. The list of streams is refreshed.

8.2.11 Exiting the AVA/ATV Manager Window

You can exit the AVA/ATV Manager Window by clicking on Quit. You are returned to the AVA/ATV Manager Browser.

8.3 Accessing the Manager Description Window

The `Manager Description Window`, shown in Figure 8.16, contains a detailed description of all managers listed in the `AVA/ATV Manager Browser`. In addition, the window consists of pull-down lists that enable you to view the available video and audio source modes and rates. For CellChains, the window displays a button for each unit.

To access the `Manager Description Window` for a particular manager, highlight the manager name in the `AVA/ATV Manager Browser` and then click on `Describe`. The `Manager Description Window` for that manager is displayed.

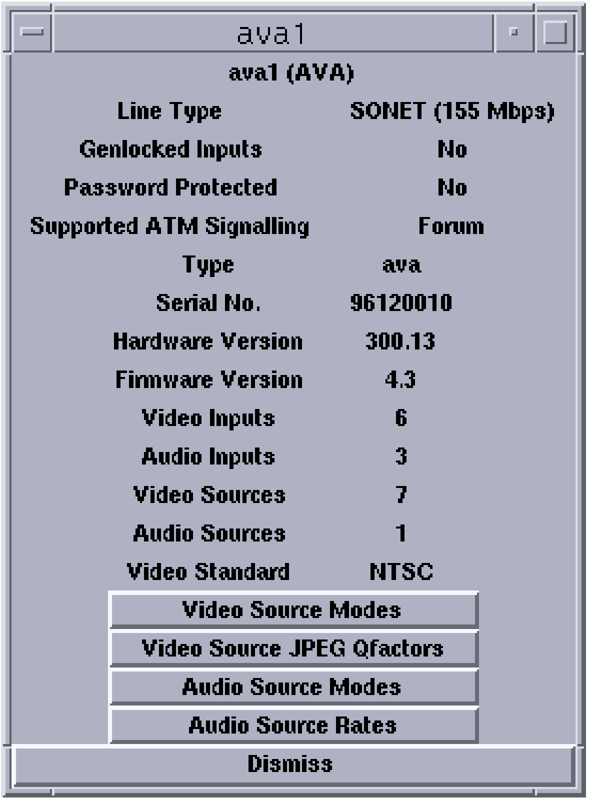

Figure 8.16 - Manager Description Window

8.4 Accessing the Trader Window

The `Trader Window`, shown in Figure 8.17, contains a list of the traders that `svc-rtds` is accessing.

Figure 8.17 - Trader Window

NOTE

The list of traders in the `Trader Window` can only be modified by restarting `svc-rtds`.

8.5 Exiting svc-rtds

To exit `svc-rtds`, click `Exit` in the `AVA/ATV Manager Browser`.

CHAPTER 9 Advanced Topics

This chapter provides information on topics of an advanced nature that you may wish to pursue after your initial video/audio ATM network is up and running. The features described in this chapter provide increased flexibility and functionality to your video network, and include:

- Netscape and Explorer Plug-in
- MBone Tools
- AVA-300 JPEG over RTP
- Multi-Homed IP Hosts
- Video and Audio PVC Streams
- Serial Line Support
- External Configuration Modules
- ATV-300 Single-Stream Mode

9.1 Netscape and Explorer Plug-in

This section describes `Netpatch`, a plug-in for Netscape Navigator and Microsoft Internet Explorer that allows you to display video streams and listen to audio streams generated by an AVA-300. Also described is how Web publishers can embed video and audio streams in their Web pages.

9.1.1 Netscape Plug-in Overview

Netscape Plug-ins are platform-dependent shared libraries. Each Netscape Plug-in registers itself with the Navigator as being able to handle one or more MIME-types. A MIME-type is a string which identifies the format of a block of data. For example, `text/html` is the MIME-type of an HTML document and `image/jpeg` is the MIME-type of a JPEG image. MIME, which stands for Multipurpose Internet Mail Extensions, was originally designed as a mechanism for embedding non-textual data in Internet mail messages.

AVA-300 streams are embedded within pages of HTML in a manner similar to that of static images, but use the `embed` tag. This tag may contain several attributes, the most important of which is the MIME-type of the object which lets the browser decide which Plug-in is to be invoked. Two other attributes are the width and height of the stream window, which are used by the browser to format the surrounding document. See the Netscape Handbook for more information on Netscape Plug-ins.

9.1.2 The Netpatch Plug-in

`Netpatch` is a plug-in that provides a wrapper around `svapatch`, allowing the integration of live AVA-300-generated audio and video streams into HTML documents. `Netpatch` is compatible with Netscape 3.0 and later, and Internet Explorer 3.0.

NOTE It is not necessary for the Web server that provides the HTML files to have ATM connectivity; only the hosts on which the Netscape client is to be run require ATM connectivity.

9.1.3 Installing the Netscape Plug-in in your Netscape Browser

To install the Netscape Plug-in in your Netscape Browser, you need the following:

- Netscape Navigator 3.0 or later or Internet Explorer 3.0, properly installed.
- SVA 5.0, properly installed (for installation instructions, see Chapter 4 or Chapter 5.

To install the Netscape Plug-in in your Netscape Browser, do the following:

1. Type the command:

```
prompt$ svarun netscape
```

   This sets the `PATH` and `LD_LIBRARY_PATH` variables to make `svapatch` available, sets the `NPX_PLUGIN_PATH` variable to make the SVA plug-ins directory known to Navigator or Explorer, and then launches the browser.

NOTE In Windows NT/95, `SETUP.EXE` automatically sets the variables during setup. For information on variable modification, refer to Section 5.2.1.

2. When the browser appears, open the URL “`about:plugins`”.

NOTE “`about:plugins`” is a program in JavaScript (a scripting language built into Navigator) that interrogates all installed plug-ins for their identification strings. You need to enable JavaScript from the `Languages` tab of the `Network Preferences` dialog in the `Options` menu.

3. Make sure the enabled field of the `Installed Plug-ins` table shows `Yes` for MIME-type `application/fore-netpatch`. Otherwise, edit the MIME configuration for `application/fore-netpatch` to show `Plug In: FORE Netpatch`. Use the `Helpers` tab of the `General Preferences` dialog, available from the `Options` menu. Reload the `about:plugins` page when you are done.

   Once the `about:plugins` page verifies that the Netpatch plug-in is enabled for MIME-type `application/fore-netpatch`, the plug-in is installed in your browser.

9.1.4 Opening AVA-300 Media Streams Inside the Netscape Browser

In this section, it is assumed that you have followed the instructions in Chapter 4 or Chapter 5 and have a managed AVA-300 called `ava1` properly configured.

NOTE It is strongly recommended that the first time you follow these instructions you use the name `ava1` for your AVA-300 manager since it is used in all of the demonstration files.

The `html` directory in the SVA distribution consists of the following set of demonstration files:

examples.html	Top-level examples file for PAL video
examples_ntsc.html	Top-level examples file for NTSC video
mono.html	`mono` video stream (PAL)
mono_ntsc.html	`mono` video stream (NTSC)
vic.html	`vic` video stream (PAL)
vic_ntsc.html	`vic` video stream (NTSC)
either.html	`jpeg-either` video stream (PAL)
either_ntsc.html	`jpeg-either` video stream (NTSC)
vat.html	`vat` audio stream (8KHz 8-bit mono μ-law)
cd.html	`cd` audio stream
vic-vat.html	`vic` video stream and `vat` audio stream (PAL)
vic-vat_ntsc.html	`vic` video stream and `vat` audio stream (NTSC)
cd-either.html	`jpeg-either` video stream and `cd` audio stream (PAL)
cd-either_ntsc.html	`jpeg-either` video stream and `cd` audio stream (NTSC)

Advanced Topics

Open the `example.html` file as a URL. The browser displays a page containing links to the Netpatch Plug-in examples in the distribution. It is a good idea to work through these by starting with the simpler examples.

NOTE System resources may limit your ability to display some video streams.

As you open each file, there will be a delay of up to several seconds while `svapatch` is loaded. Once `svapatch` has started, you will see the animated video and/or hear the audio from the streams embedded within the page. When you wish to close the connection to the AVA-300, navigate to a different location by clicking the `Back` button or close down the Netscape browser. If you resize the browser while it is displaying a page containing embedded video streams, you will experience a delay as `svapatch` is reloaded.

If you specify a non-zero screen area for your embedded object, diagnostic messages appear in the Navigator's status bar whenever you point your mouse within the area. Therefore, it is advisable to allocate a non-zero width and height even for audio plugins as this allows you to monitor `netpatch's` progress and diagnose problems more easily.

Status messages appear to warn you if `netpatch` is unable to locate the specified manager, video or audio stream, or if ATM appears to be disconnected. Exceptional conditions that occur after the connection has been successfully established, such as the resetting of the manager, are generally unrecoverable and will be announced with the message `error`. If this occurs repeatedly, you should try invoking `svapatch` from the command line to test connectivity and stream/sink configuration (see Section 7.8).

9.1.5 Embedding Video and Audio Streams in HTML Documents

Netpatch allows users of the same ATM video network to create documents containing video and audio streams which can be shared on a Web site, sent to each other via e-mail, and bookmarked for later use, using exactly the same techniques as with other media on the Web.

Creating pages with embedded streams is no more difficult than creating pages with static images. The only requirement is that your intended audience must have ATM connectivity to the managed device generating the streams. This section explains the `embed` tag and the attributes that Netpatch uses. An understanding of HTML and web authoring are assumed.

An HTML `embed` tag takes the following form:

```
<embed arg=value ...>
```

where `arg` is the name of the attribute and `value` is the value assigned to it.

The following general points apply to the `embed` tag:

- To leave an attribute at its default value, omit it altogether.
- To specify an attribute as having the null string value, use `arg=""`.
- You may not specify an attribute without a value, e.g., `arg=`.
- Put quotation marks (“”) around attributes containing spaces.
- All embedded objects must have at least the first three of the attributes listed below (`type`, `width`, and `height`). All other attributes are passed to the Plug-in.

The following is an example of the contents of an HTML file that displays a video stream in your Netscape browser:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<title>Netpatch: example</title>
</head>
<embed type=application/fore-netpatch
manager=ava1
trader=10.11.3.30
videostream=rgb16
audiostream=cd
width=304
height=224>
</body>
</html>
```

The following attributes can be specified in an embed tag in any page of HTML which is to activate the Netscape Plug-in:

type
Sets the MIME type of the embedded object. The value must be `application/fore-netpatch`.

width
height
Sets the size of the embedded object, thus allowing the browser to format the document properly. If the size you specify is too small for the video stream, the image is clipped to fit. If you specify either of these as zero, the hidden attribute is implicitly specified (see the following entry). Since this can cause problems with Netscape 3.0, give all embedded objects a non-zero width and height.

hidden=true
Specifies that the embedded object occupies no screen area. This tag is only appropriate for audio streams since video streams require window creation in order to be displayed. Since hidden plug-ins are known to cause problems with Netscape 3.0, you should avoid using this attribute.

trader
Specifies a comma-separated list of hosts on which SVA traders are running. The default value is `'localhost'`. This attribute is not case-sensitive since it is the DNS name of an Internet host.

manager
Specifies the managed AVA-300 from where you wish to export a stream. This attribute has no default value and is mandatory.

videostream
audiostream
Specify the names of the video and audio stream(s) that you wish to attach to. If you do not assign either attribute a value, the respective medium (video or audio) is ignored. Neither attribute has a default value.

videosink
audiosink
Specify the names of the video and audio sinks that you wish the Plug-in to use. If either attribute has the null string value (default) or the specified sink cannot render the corresponding stream, `svapatch` picks the most appropriate sink for the stream type, depending on the workstation‘s capabilities. It is not recommended that you specify sinks yourself, since all platforms may not be capable of using the specified sink. To obtain a list of sinks supported by the workstation, type:

`svarun svapatch -show_local_sinks`

vwidth
vheight

Specify the actual, unscaled size of the video stream. These are optional, and if both are specified are used in preference to the `width` and `height` attributes when deciding which sink to use (see notes regarding the Matrox JPEG sink below). This allows you to change the width and height of the video window arbitrarily without affecting the choice of sink.

volume

Specifies the initial audio volume. The default value is `15`. The acceptable range is `0-31`. This attribute sets the volume for all browsers. A better solution is to use the standard audio control tool to set the workstation's global audio device volume dynamically. (`apanel(1)` on IRIX and `audiocontrol(1)` on Solaris).

audioout

Specifies the workstation audio output port to use. Acceptable values are `headphones`, `speaker`, or `lineout`. The default is `headphones`. Some platforms do not support all three values; for example, IRIX ignores the setting—you remove the headphones from the socket to play through the speaker. You can use standard workstation or PC tools to control this dynamically.

videoscaling

Instructs the Plug-in to double the video vertically or both vertically and horizontally. The default is `small`. Possible values include `small`, `double_y`, or `double_xy`. If you specify doubling, you should adjust the `width` and `height` fields (see above) to correspond to the doubled size. Videoscaling is useful for scaling single field video up to full frame size.

rate_match

Sets whether or not audio rate matching is to be enabled. Specify `0` to disable, `1` (default) to enable.

audio_playout_level
audio_playout_max

Specify in milliseconds the target and maximum audio playout latencies. The defaults are `400` and `600`, respectively.

NOTE

If you are using a Sun machine, and the `embed` tag's `videosink` attribute is unspecified, the Plug-in will use a videosink setting of `parallax`. This causes the Parallax sink to be used if a Parallax JPEG decompression card is installed. The video will be unscaled (regardless of the value of the `videoscaling` attribute).

NOTE

If you are using a PC (running Windows NT or 95) and the `embed` tag's `videosink` attribute is unspecified, if `vwidth` and `vheight` specify one of the sizes listed below, or if `vwidth` or `vheight` are undefined and `width` and `height` correspond to a scaling (corresponding to the videoscaling attribute) of one of the sizes listed below, then the Plug-in will use a videosink setting of `active_jpeg`. This causes the RainbowRunner JPEG sink to be used if a Matrox RainbowRunner MJPEG decompression card is installed. The video stream will be scaled according to the `videoscaling` attribute, up to the limits imposed by the RainbowRunner (which depend on screen mode and depth).

The sizes accepted by the `active_jpeg` sink are a width of `704, 352, 176,` or `640`, and a height of `288, 144, 240,` or `120`. The first two heights will cause the video output of the RainbowRunner card to produce PAL video, the last two NTSC.

NOTE

Non-zero `width` and `height` attributes are required for the Plug-in to be initialized.

NOTE

The `audioout` attribute has no effect on IRIX—you must insert/remove headphones into/from the headphone socket as desired.

NOTE

You need to change the manager and possibly the `audiostream` and `videostream` names to use these examples in your Web pages.

NOTE

Illegal characters in any of the attributes will cause `netpatch` to abort with an error message. This is a security measure designed to prevent HTML that contains embedded objects in which arbitrary shell commands are concealed.

9.2 MBone Tools

The Internet community has developed a video and audio distribution system which is implemented over an IP multicast network called the Multicast Backbone (MBone). This network has a wide geographic coverage that allows interaction between many sites worldwide. Public domain tools that are compatible with most workstations are available for video and audio distribution and conferencing on the MBone.

This chapter describes the MBone tools supplied with the SVA software, which consist of modified versions of the public domain tools `vic` and `vat`, originally developed by the Network Research Group at the Lawrence Berkeley National Laboratory (LBNL).

NOTE See `http://www-nrg.ee.lbl.gov` for more details on `vic` and `vat`.

`vic` and `vat` use RTP (Real-time Transport Protocol) to frame the multimedia payload. Using these modified tools, you can route video and audio from AVA-300s onto the MBone. The default AVA-300 streams include `vic` and `vat`, which are suitable formats for these tools.

You need MBone connectivity, which consists of a multicast-capable workstation and a multicast route to the MBone, to participate in conferences or watch events on the MBone. Multicast IP support is available in the base release of many popular workstation operating systems (e.g., Sun Solaris 2.X and IRIX 6.3) and network router implementations. The tools may also be used on private multicast IP networks or in standard point-to-point IP mode.

9.2.1 vic

`vic` (Video Conference) is a tool that allows you to participate in video conferences or watch broadcast events. vic can operate in one of two modes:

Point-to-point mode Conferencing between two fixed participants.

Multicast mode Conferencing between an arbitrary number of participants.

vic supports a number of video grabber devices which are capable of capturing video from different vendors' workstation video devices (e.g., Parallax, SGI Galileo, or DEC J300). The SVA modifications to `vic` add an AVA-300 grabber module that allows AVA-300 video to be transferred from the SVA application `svapatch` to `vic`, just as if the video was originating from a local video grabber device. See Section 9.3 for issues regarding AVA JPEG and RTP.

NOTE The AVA-300 grabber is compatible only with JPEG format video.

Advanced Topics

9.2.2 Running svapatch for vic

Before setting up a patch for use with `vic`, ensure that the traders and managers can be contacted successfully. Do this by patching AVA-300 video to the workstation or PC display using **`-local svapatch`**, as in this example:

```
svarun svapatch -from ava1 -to -local -video vic
```

The `vic` video stream from AVA-300 `ava1` is displayed on the workstation / PC.

NOTE If you have problems, or if the video stream is not displayed, please refer to Chapter 7 for further information.

Terminate `svapatch` and then start a `vicsink` patch to act as video input to `vic`:

```
svarun svapatch -from ava1 ‘vicsink vic‘
```

The `‘vicsink vic‘` portion of the command executes a helper script to construct appropriate arguments to `svapatch`.

The following message is displayed at periodic intervals:

```
.... no consumer running
```

This message indicates that `svapatch` is waiting for you to start a `vic` process to which it can send the incoming AVA-300 video.

NOTE The `vicsink` helper script is only available on UNIX. On Windows NT, the full command line must be entered:

```
svarun svapatch -from ava1 -to -local -video vic -args jpeg_null \
  -vic_2_6_fix 1 -record_start NamedShm Raw //./pipe/vic_named \
  -endargs
```

NOTE `vic` on Windows NT requires a different named pipe than on UNIX. Running `svapatch` for `vic` is not supported on Windows 95.

9.2.3 Transmitting AVA-300 Video with vic

To start a `vic` process to transmit video, do the following:

1. Open a window on the workstation / PC.
2. At the prompt, type:

   svarun vic *<IP address>***/***<UDP port>*

3. To run in point-to-point mode (recommended for initial testing), enter a standard hostname or IP address and an available User Datagram Protocol (UDP) port:

   svarun vic phoenix/8080

4. To run in multicast mode, enter a class D multicast IP address.

NOTE Two ports are actually used (*<UDP port>* and *<UDP port>+1*) are required. One port is for data and the other is for control.

The main `vic Window`, shown in Figure 9.1, is displayed.

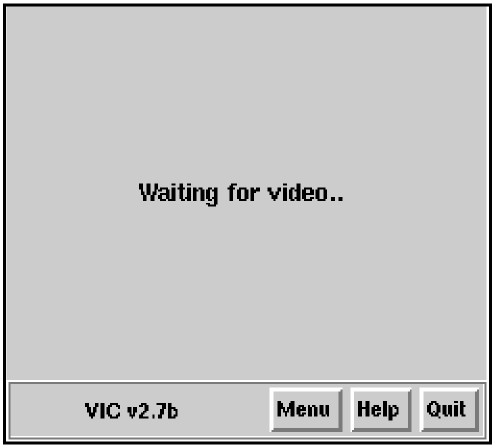

Figure 9.1 - vic Window

5. Click on the `Menu` button.

   The `Transmission Control Window`, shown in Figure 9.2, is displayed.

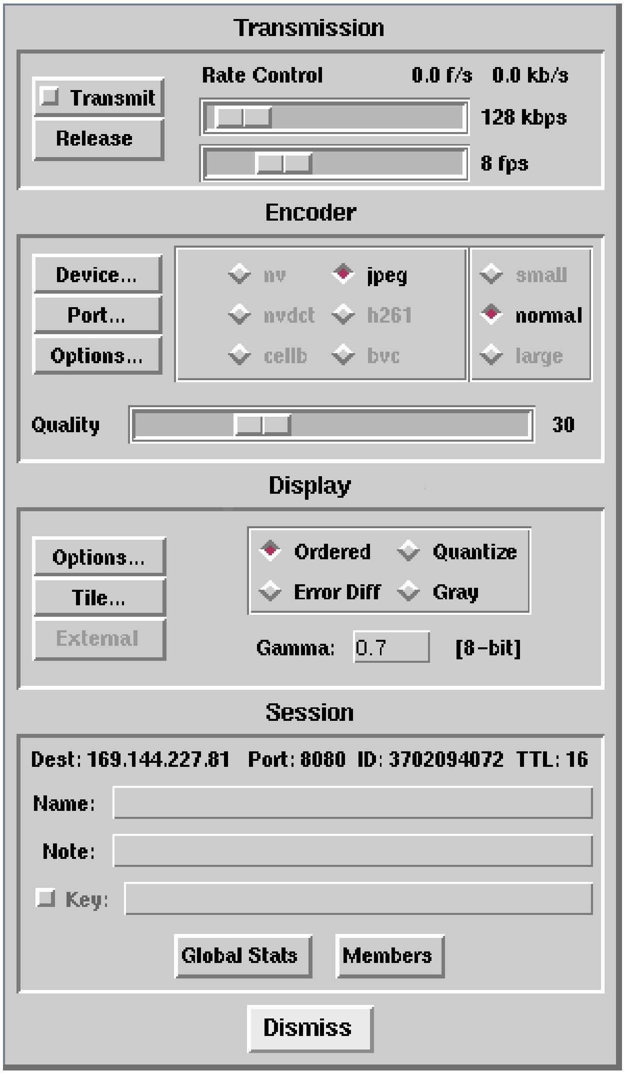

Figure 9.2 - Transmission Control Window

6. Click on the `Device` button.

   A menu containing available grabbers, as shown in Figure 9.3, is displayed.

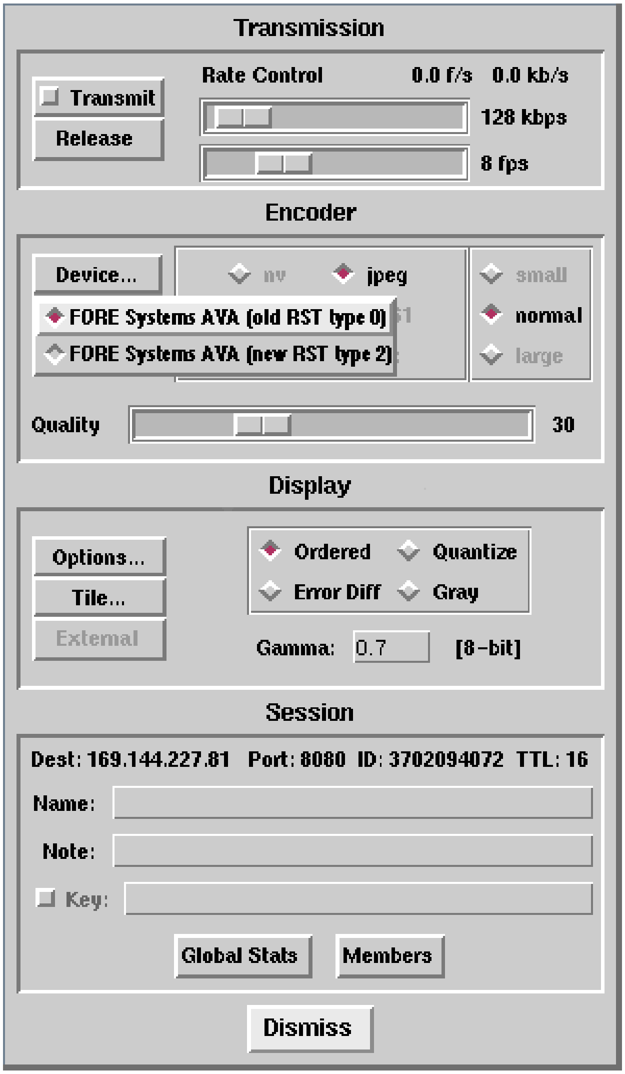

Figure 9.3 - Available Grabbers Menu

NOTE

In addition to a list of supported video hardware you have on your workstation, you should see `FORE Systems AVA (old RST type 0)` and `FORE Systems AVA (new RST type 2)`. If they are not listed, make sure that you do not have another version of `vic` on your path before the SVA version. See Section 9.3 for a discussion of the RST type options.

7. Double-click on `FORE Systems AVA (old RST type 0)` or highlight it with a single click and then click on the `Transmit` button.

   A thumbnail video image is displayed in the first `vic` window. The video is being transmitted to the destination workstation or multicast group.

NOTE

The `Rate Control` slider can be used to limit the amount of data transmitted, and causes `vic` to drop frames if it is set lower than the rate coming out of the AVA-300. If you wish to use a low frame rate, you should use a low rate stream from the AVA-300 to avoid overloading the workstation.

NOTE

If you see continuous `... buffers full ...` messages from the `svapatch` process, this is an indication that `vic` is unable to consume (or drop) the frames from `svapatch` quickly enough. You should use a lower rate AVA-300 stream.

To change the rate from the AVA-300, you can either choose a different stream or edit an existing one with `svc-rtds`. Experiment with different sizes, frame rates, and Q-factors, and choose the one most appropriate for your needs. For example, you may require fewer high-quality frames or more lower-quality ones.

NOTE

Any available supported workstation capture devices can still be used by selecting them from the `Device` menu.

9.2.4 Receiving AVA-300 Video from vic

To receive video from a point-to-point session, type:

svarun vic *<source IP address>***/***<UDP port>*

Where *<source IP address>* is the IP address of the workstation/PC transmitting the video, and *<UDP port>* corresponds to the port specified by the transmitting `vic`.

NOTE In a point-to-point session, the transmitter and receiver must specify the same UDP ports.

NOTE While only two parties may participate in a point-to-point video session, both may participate in several such sessions.

To receive video from a multicast session, use the address and port of a currently-available session.

NOTE The sd (session directory) tool, available from LBL, can be used to find the multicast addresses of currently-available sessions.

The main `vic Window` is displayed with a thumbnail of the available video stream. Click on the image to see it full-size.

NOTE See the supplied `vic.1` manual page for more details.

9.2.5 vat

`vat` is similar to `vic`, but sends and receives audio rather than video. The SVA distribution includes a version of `vat` that accepts AVA-300 audio from `svapatch`.

NOTE

Although the AVA-300 and `vat` support a number of different audio formats, only 8-bit mono μ-Law audio from an AVA-300 may be used with `vat`.

9.2.6 Running svapatch for vat

Before setting up a patch for use with `vat`, you should make sure that the traders and managers can be contacted successfully. You do this by patching AVA-300 audio to your workstation / PC:

```
svarun svapatch -from ava1 -to -local -audio vat -args -speaker\
-volume 10 -endargs
```

The `vat` audio stream from AVA-300 `ava1` is played on the workstation / PC.

Terminate `svapatch` and then start a `vatsink` patch to act as audio input to `vat`:

```
svarun svapatch -from ava1 ‘vatsink vat‘
```

The `‘vatsink vat‘` portion of the command executes a helper script to construct appropriate arguments to `svapatch`.

The following message is displayed at periodic intervals:

```
.... no consumer running
```

This message indicates that `svapatch` is waiting for you to start a `vat` process to which it can send the incoming AVA-300 audio.

NOTE

The `vatsink` helper script is only available on UNIX. On Windows NT, the full command line must be entered:

```
svarun svapatch -from ava1 -to -local -audio vat -args audio_null \
-record_start NamedRaw Raw //./pipe/vat_named -endargs
```

NOTE

`vat` on Windows NT requires a different named pipe than on UNIX. Running `svapatch` for `vat` is not supported on Windows 95.

9.2.7 Transmitting AVA-300 Audio with vat

To start a `vat` process to transmit audio, do the following:

1. Open a window on the workstation / PC.
2. On UNIX at the prompt, type:

   svarun vat -U /tmp/vat_named *<IP address>***/***<UDP port>*

   The **-U** argument instructs `vat` to use the named pipe **/tmp/vat_named** as its audio input device. This corresponds to the pipe which is set up by the `svapatch` process.

   On Windows NT, the pipe **//./pipe/vat_named** is used instead.

   The main `vat Window`, shown in Figure 9.4, is displayed.

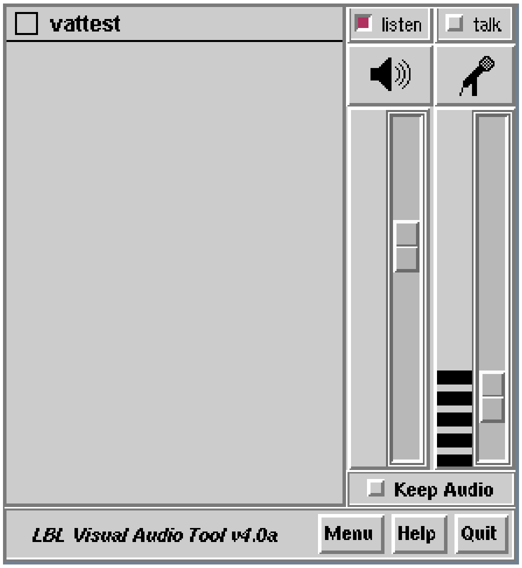

Figure 9.4 - vat Window

3. Click on the `Talk` button to start transmitting. The displayed volume meter bar visually represents changes in the transmitted audio level.

9.2.8 Receiving AVA-300 Audio with vat

To receive audio, type:

svarun vat *<source IP address>***/***<UDP port>*

NOTE

vat assumes that the same audio device used for output is to be used for input, so you cannot play received audio to the local workstation or PC while transmitting AVA-300 audio.

NOTE

As with vic, two UDP ports are actually required. See Section 9.2.3.

NOTE

See the supplied vat.1 manual page for more details.

9.3 AVA-300 JPEG Over RTP

The Internet Engineering Task Force (IETF) Audio/Video Transport Workinggroup (AVT) (http://ietf.org) has submitted an RFC (RFC 2035), defining the payload format for JPEG compressed video over RTP.

Six types of JPEG payload (0-5) are defined. The odd types differ from the even only in sampling factors. AVA-300 video is 4:2:2 JPEG, corresponding to the even types.

AVA-300 JPEG contains restart markers. The original JPEG payload types (0/1) did not allow for the presence of restart markers, so types 2-4 were added. AVA-300 JPEG should be framed using type 2. However, most existing RTP video tools (such as `vic`) do not yet support types other than 0/1. The SVA distributed `vic` supports all six types.

The SVA modified `vic` can send JPEG either as type 0 or type 2. Type 2 is the legitimate format, but at present the only known tool which can receive it is SVA modified `vic`. It can, however, also send type 0. This is contrary to the payload type, since the JPEG contains restart markers, but the transmitted video is modified in such a way that it can be accepted by the plain LBL distribution of `vic`. Other tools are not likely to accept type 0 with restarts; therefore, type 0 should not be used for transmission onto multicast networks where you cannot guarantee what tools will be used for viewing.

9.4 Multi-Homed IP Hosts

On many host computers, the ATM network interface card is additional to a motherboard-based Ethernet interface. In some environments, if both interfaces are assigned IP addresses it is possible to have complex IP configurations. For example, it may not be possible to route IP packets between the two subnets to which the host is connected.

Many SVA applications use IP for management and control purposes and so the structure and connectivity of the local network is important. Unless otherwise directed, SVA applications will send and receive packets on the default IP interface.

You may force the SVA software to use a particular network interface by using the `-localhost` option to specify the IP address of the desired network interface.

NOTE It should not be necessary to use the SVA `localhost` facility on most networks.

For example, consider a multi-homed host which is attached to the ATM network using address `1.1.1.61` and also to the Ethernet using address `192.168.80.61`. The default IP address is that for the Ethernet and there is no routing between the two subnets.

To start a trader on the host which utilizes the ATM interface, type the following:

```
svarun trader -localhost 1.1.1.61
```

To start a manager for an AVA-300 device attached to port `1C2` on switch `orion`, type the following:

```
svarun -device orion:1c2 -localhost 1.1.1.61 -trader 1.1.1.61
```

The manager will register itself with the local trader and accept management requests from the ATM network.

To start an `svc-rtds` application on the host which is able to access the above trader and manager, type the following:

```
svc-rtds -localhost 1.1.1.61 -trader 1.1.1.61
```

To start another `svc-rtds` application on a similarly multi-homed host on the same subnets and with IP address `1.1.1.62` on the ATM network, type the following:

```
svc-rtds -localhost 1.1.1.62 -trader 1.1.1.61
```

The `svc-rtds` application will contact the trader on host `1.1.1.61` using IP packets over the ATM network. The manager for the AVA attached to port `1C2` on switch `orion` should be accessible.

9.5 Video and Audio PVC Streams

Although SVCs are the preferred means of carrying video and audio data, there are certain circumstances when their use is not possible nor desired. For example, certain ATM switches do not yet support SVCs, or you may wish to direct video and audio streams to a non-SVA-compatible device or software application which does not fully support SVCs. To address these situations, the SVA software allows you to use PVCs to carry video and audio streams from an AVA-300 to a client application, ATV-300, or any other application or device.

All of the examples used in this section assume that the physical configuration described in Chapter 4 or Chapter 5 is used:

- The AVA-300 is attached to port `1A1` on switch `orion`.
- The ATV-300 is attached to port `1B1` on switch `orion`.

In order to work through the examples described in this chapter, a number of PVCs must be created. These PVCs are used to transport video and audio from the AVA-300 to the ATV-300.

To configure the video and audio PVCs, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Configure the audio and video PVCs from the AVA-300 to the ATV-300, as follows:

```
new 1A1 0 150 1B1 0 150

new 1A1 0 151 1B1 0 151
```

NOTE The PVCs are unidirectional since the data needs only to flow from the AVA-300 to the ATV-300.

9.5.1 Non-Managed Video/Audio PVCs

The SVA utilities `avaconfig` and `atvconfig` are used to configure the AVA-300 and ATV-300 respectively to send and receive video and audio streams over PVCs.

NOTE `avaconfig` and `atvconfig` do not function if the referenced device is being managed.

NOTE `avaconfig` and `atvconfig` do not function in a CellChain environment. In order to control CellChain systems, the `svamgr` program must be used.

NOTE The AVA-300 and ATV-300 do not remember `avaconfig` or `atvconfig` configurations if power-cycled or reset. To create permanent PVC configurations, you must either use an External Configuration Module (see Section 9.7 or use managed PVC stream definitions (see Section 9.5.2).

The example in the following section configures the AVA-300 to send one video and one audio stream on the PVCs created in Section 9.5 through to the ATV-300.

9.5.1.1 avaconfig

To configure the AVA-300 to send both a video and audio stream on the PVCs created, type the following:

```
svarun avaconfig -device orion:1A1 -video source atv vci 150 \
    -audio source atv vci 151
```

The AVA-300 transmits a video stream on VCI 150 and an audio stream on VCI 151, which are the PVCs configured through to the ATV-300. The ATV-300 does not output analog video and audio at this point since it has not been configured with the details of the incoming streams.

NOTE The video and audio streams are assigned separate VCIs. Failing to do so causes the cells for different AAL5 PDUs to become interleaved and create reception errors at the ATV-300.

NOTE

Additional video and audio transmission/processing options are described in the `AvaVideoAttrParseArgs` and `AvaAudioAttrParseArgs` manual pages located in Appendix A.

9.5.1.2 atvconfig

To configure the ATV-300 to receive and process the video and audio streams from the AVA-300, type the following:

```
svarun atvconfig -device orion:1B1 -video source atv vci 150 \
                   -audio source atv vci 151
```

NOTE

Additional video and audio reception/processing options are described in the `AtvVideoAttrParseArgs` and `AtvAudioAttrParseArgs` manual pages located in Appendix A.

9.5.2 Managed Video/Audio PVCs

It is possible to create PVC stream definitions in managed AVA-300 and ATV-300 devices. The advantage of doing so over using `avaconfig` and `atvconfig` is that the manager is able to maintain the AVA-300 and ATV-300 configuration across device reset conditions. Also, managed devices are capable of handling PVC and SVC streams concurrently.

If an AVA-300 is currently being managed, then it is possible to use `svc-rtds` to create/edit PVC audio and video streams. Chapter 8 describes how to create and edit SVC streams. The `SVCs/PVCs Enabled` button on the `Video Edit Window` and `Audio Edit Window` (see Section 8.2.6 and Section 8.2.7) may be toggled to `PVCs Enabled` to create a PVC stream.

An alternative to using `svc-rtds` is to directly edit the manager configuration file to define PVC streams. This must be done for ATV-300 systems since `svc-rtds` does not have the facility to create and edit sink stream definitions.

NOTE

The manager process should not be running while its configuration file is being edited.

To create the PVC configuration detailed in Section 9.5.1.1, add the following to the AVA-300 manager configuration file:

```
video
{
name videoinputpvc
source atv
vci 150
}

audio
{
name audioinputpvc
source atv
vci 151
}
```

To create the PVC configuration detailed in Section 9.5.1.2, add the following to the ATV-300 manager configuration file:

```
video
{
name videooutputpvc
source atv
vci 150
}

audio
{
name audiooutputpvc
source atv
vci 151
}
```

NOTE Streams in managed devices must be assigned per-device unique names.

NOTE The format of the manager configuration file is further detailed in the `svadefaults` manual page located in Appendix A.

When the AVA-300 and ATV-300 device managers are re-started, the above PVC configurations are activated. The ATV-300 receives and processes the audio and video PVC streams from the AVA-300.

9.6 Serial Line Support

The AVA-300 has a bi-directional serial line (RS-232) interface which may be used to transmit and receive serial data over the ATM network.

NOTE

The ATV-300 also has a serial line interface, but it is not used at this time.

The SVA architecture defines both serial stream sources and serial stream sinks in a manner similar to that used for video and audio traffic. The AVA-300 can act as both a serial stream source and a serial stream sink.

NOTE

The AVA-300 serial hardware operates at 8 bits data plus 1 stop bit with no parity. Hardware flow-control schemes (e.g., RTS/CTS) are not supported.

WARNING!

Care must be taken when using the serial interface on AVA-300 systems at hardware versions earlier than 300.10. The hardware version of your AVA-300 may be ascertained using the `avareset` program. Please refer to Section 4.5. In particular, using a serial cable with pins 4 (DTR) and 8 (CTS) connected from the AVA-300 (Data Set end) to the remote device (Data Terminal end) may result in permanent damage to the AVA-300 serial port hardware. Using a cable with DTR directly connected to CTS at the AVA-300 end will not cause any problems. Under normal circumstances, a cross-over (Null Modem) cable with only pins 2 (Rx), 3 (Tx) and 5 (Ground) connected will work without any problems. However, the only safe option is to ensure that pins 4 and 8 are not connected.

Serial data is transmitted on the ATM network in single cell AAL5 packets.The format of the 40-byte data bytes in an AVA-300 serial packet are as follows:

sequence number	4 bytes (not used)
opcode	1 byte set to 0x10
length	1 byte: number of valid serial bytes below
data	32 bytes of serial characters
pad	2 bytes set to zero

The AVA-300 may be configured to transmit serial characters on to the ATM network in a variety of configurations. The main parameters that may be configured are:

Baud Rate The AVA-300 serial hardware supports a variety of baud rates: `300, 600, 1200, 2400, 4800` and `9600` baud. The default baud rate is `9600`.

Count The AVA-300 can be configured to buffer a certain number of serial characters internally prior to transmission on to the ATM network. The maximum (and default) count is `32` characters (see packet format above).

Delay The AVA-300 can be configured to buffer serial data for a certain amount of time prior to transmission on to the ATM network. The maximum delay that can be set is `4000` milliseconds. The default serial input delay is `80` milliseconds.

NOTE See the `AvaSerialAttrParseArgs` manual page in Appendix A for a full description of all the parameters that can be associated with serial source and sink streams.

NOTE The AVA-300 is not capable of supporting independent baud rates in the input and output direction.

9.6.1 Serial Data SVCs

Serial source streams may be defined in SVA manager configuration files in a manner similar to those used for video and audio streams. The default manager configuration file `svadefaults` contains a definition for a single serial source stream which is called `serial`. This stream has the default characteristics for serial input streams described in Section 9.5. The stream is listed in the `AVA/ATV Manager Window` as illustrated in Figure 4.4.

NOTE

`svc-rtds` does not allow serial data to be received via the GUI, nor does it allow serial stream definitions to be created/edited.

`svapatch` may be used to transmit the serial source data from one AVA-300 to be sent over an SVC to a second AVA-300 where it is output on the serial port of that unit. This is illustrated in the following example:

```
svarun svapatch -from ava1 -to ava2 serial
```

A single serial input may be multicast so that it is output via many AVA-300s located on different switch ports in the ATM network.

NOTE

It is not possible for a CellChain system to terminate the same incoming serial multicast SVC on separate AVA-300s in the chain.

9.6.2 Serial Data PVCs

This section discusses creating PVC serial streams in non-managed and managed AVA-300s.

9.6.2.1 Non-Managed Serial PVCs

The `avaconfig` command may be used to configure both PVC serial source and sink operation for an AVA-300. To demonstrate the operation of a single AVA-300 transmitting and receiving serial data, a *ForeRunner* switch may be used to loop back transmitted cells and send them back to the originating AVA-300.

To configure a non-managed PVC, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Type the following at the prompt:

```
new 1A1 0 190 1A1 0 191
```

6. To transmit and receive serial data, configure the AVA-300 with the following two stream definitions:

```
svarun avaconfig -device orion:1A1 -serialin vci 190 -serialout vci
191
```

This configures the AVA-300 to transmit serial data cells using VCI 190 to the ATM switch, which then translates the VCI to 191 and sends the cells back to the AVA-300. Since the AVA-300 has been configured to receive serial data cells from the ATM network on VCI 191 and output them on its serial port, typing any characters at the AVA-300 serial port will cause them to be echoed.

By using multiple AVA-300 devices, you can set up serial stream connections across the ATM network. You can also use software on workstations or PCs to transmit and receive serial data from AVA-300 devices. This requires you to use a native ATM API such as XTI on UNIX or Winsock2 on Windows platforms. In both cases, the native ATM PVC interface is used and the programmer exchanges packets in the AVA-300 serial data format with the remote device.

9.6.2.2 Managed Serial PVCs

It is possible to create PVC serial streams in a managed AVA-300 device. In order to do this, you must edit the manager configuration files.

NOTE

The manager process should not be running while its configuration file is being edited.

To create the PVC configuration detailed in Section 9.5.1.1, add the following to the manager configuration file:

```
serialin
{
name serialinputpvc
vpi 0
vci 190
}

serialout
{
name serialoutputpvc
vpi 0
vci 191
}
```

NOTE

The format of the manager configuration file is further detailed in the `svadefaults` manual page located in Appendix A.

9.7 External Configuration Modules

External Configuration Modules (ECMs) are used to store pre-canned PVC stream configurations. An AVA-300 or ATV-300 with an active ECM plugged in takes its configuration from the ECM when it is powered on. ECMs may be programmed remotely over the ATM network or additionally, in the case of the ATV-300, by using the GUI. See Section 9.8.5.

NOTE The ECM configurations detailed in this section assume the PVC configuration outlined in Section 9.5.

9.7.1 Installing ECMs in the AVA-300 and ATV-300

To install ECMs in the AVA-300 and ATV-300, do the following:

1. Power-down the AVA-300 and ATV-300.
1. Plug an ECM into the AUX DIN Connector on the rear panel of the AVA-300.
2. Plug an ECM into the AUX DIN Connector on the rear panel of the ATV-300.
3. Power-up the AVA-300 and the ATV-300.
4. Reset the AVA-300 and ATV-300:

```
svarun avareset -device orion:1A1

svarun atvreset -device orion:1B1
```

NOTE In addition to the firmware version information, you will receive verification that the ECMs are installed.

5. Configure the ECMs by typing:

```
svarun ecmconfig -set -device orion:1A1 -video source atv vci 150\
                    -audio source atv vci 151

svarun atvconfig -ecm -device orion:1B1 -video source atv vci 150\
                    -audio source atv vci 151
```

NOTE

Resetting the AVA-300 and ATV-300 over the network results in their performing as though the ECMs are not present.

6. To cause the AVA-300 and ATV-300 to begin executing their ECM configurations, type the following:

```
svarun avareset -ecm -device orion:1A1

svarun atvreset -ecm -device orion:1B1
```

NOTE

A variety of configurations may be used to configure stream definitions. Refer to the `ecmconfig` and `atvconfig` manual pages located in Appendix A for additional information.

Advanced Topics

9.8 ATV-300 Single-Stream Mode

`Single-stream mode` is provided as a simple method of creating one JPEG video stream and one audio stream from a specified AVA-300 to an ATV-300. Once you have established a connection, you may alter video stream characteristics such as the AVA-300 input channel being used, JPEG Q-factor, and frame rate.

Two varieties of single-stream mode are supported by the ATV-300:

Master mode Provides you with full control over the stream characteristics and causes reconfiguration of the AVA-300 when necessary.

Slave mode Assumes that another ATV-300 in master mode has configured the AVA-300 to send the video and audio streams over multicast PVCs, and simply configures the ATV-300 to receive those streams. The ATV-300 then dynamically adapts to changes in the stream characteristics caused by the master ATV-300.

NOTE Only one ATV-300 should be running in master mode against an AVA-300 at any one time. The number of slave mode ATV-300s which can receive the resulting streams is limited only by the number of multicast PVCs available.

9.8.1 Single-Stream Mode Initialization

To initialize Single-stream mode, do the following:

1. Select `Single-Stream Mode` from the ATV-300 `Main Menu`.
2. Specify the video standard being used at the AVA-300 inputs.

NOTE This need not be the same as the current ATV-300 video output mode.

Selecting `PAL` or `NTSC` causes `master mode` to be selected. You also have the option to select `Slave mode only`.

3. If in `master mode`, specify the VCIs for the control, video, and audio streams between the remote AVA-300 and the ATV-300, all of which must be distinct.

NOTE

If you press the `menu` button at any point before the audio VCI is confirmed, the operation is abandoned and you are returned to the ATV-300 `Main Menu`.

NOTE

When you select `master single-stream mode`, the specified control channel must be bidirectional, though the video and audio channels may only be from the AVA-300 to the ATV-300. Additionally, the VCIs for all three streams must be symmetrical between the ATV-300 and the AVA-300, that is, they must not be re-mapped to different values by the ATM network, and the PVCs' VPIs must all be `0`.

NOTE

In `slave mode`, the VCIs given for the video and audio streams specify two PVCs onto which the video and audio streams from the AVA-300 are being multicast (usually by a switch in the ATM network) to arrive at the ATV-300. The only restriction on those PVCs is that their VPIs must again be equal to `0`.

9.8.2 Master Mode Initialization

If `master mode` is being used, a menu showing the default AVA-300 configuration is displayed, along with options to change the video and audio input configurations (between composite and S-Video and between mic and line level, respectively), to set the audio gain, and to adjust the peak data rate control on the video stream. The AVA-300 version and network interface type are noted to determine which video inputs are configurable and the maximum available data rate from the unit.

NOTE

Only audio input `1` may be switched between mic and line level. Inputs `2` and `3` expect line level signals only.

When the correct configuration has been set up, selecting the first menu item sends it to the remote AVA-300. The best quality compressed video that the remote AVA-300 supports is used. If supported, this is Variable Q-Factor; if not, it is Q-Factor 20. The video stream is sampled from input 1; the audio stream is also sampled from input 1 at 48KHz.

NOTE

If the ATV-300 is outputting PAL video and the AVA-300 is sampling NTSC, the video stream is centered on the display. Conversely, if the ATV-300 is outputting NTSC and the AVA-300 is sampling PAL, the AVA-300 is made to scale the stream to fit an NTSC display before sending it to the ATV-300. In either of these cases, the video may jerk every few seconds. This is normal and is due to the differing frame rates at the video source and the ATV-300's output.

9.8.3 Master Mode Operation

Pressing the `menu` button while in `master single-stream mode` brings up a different menu from the usual ATV-300 `Main Menu`, allowing you to change the Q-factor and frame rate of the video stream, and the video and audio channels from which the streams are being sampled. You may also save the current stream settings to an ECM, if one is attached, and the current user configuration, change the VCIs being used in order to communicate with a different AVA-300, and exit single-stream mode to return to normal ATV-300 operation. It is not possible to switch directly from `master mode` to `slave mode` or vice-versa; in order to do this, you must select the `Leave Single-Stream Mode` option and then re-enter `single-stream mode` from the ATV-300 `Main Menu`.

If an AVA-300 is being used, an option to toggle interlacing on and off on the video stream is provided and a `Picture Control` option yields a menu allowing you to alter various characteristics of the video stream, including brightness, contrast, color saturation and hue of the video output and the volume of the audio output. The color of the video picture may also be inverted, the stream horizontally reflected, and the state of the text banner at the bottom of the display may be changed. A further option on this menu resets all video output settings and the audio volume to their default levels.

On the Remote, buttons `1-6` may be used to change the video and audio input being sampled. The audio inputs are paired with the video inputs such that selecting video input `1`, `2` or `3` selects audio input `1`, `2` or `3` respectively. Selecting video input `4`, `5`, or `6` also selects audio input `1`, `2`, or `3`, respectively. If two AVA-300 video inputs were set up to be used as a single S-Video input at the configuration stage, pressing the higher channel button has no effect; for example, if inputs `2` and `5` are configured as S-Video, pressing button `2` selects that input and pressing button `5` has no effect.

9.8.4 Slave Mode Operation

`Slave mode` is a more limited form of `single-stream mode` allowing the ATV-300 to "listen in" to streams being controlled by another ATV-300 running in `master mode`. While the ATV-300 is in `slave mode` it continuously adapts to changes in the stream configuration caused by the controlling master ATV-300, such as changes in Q-factor, interlace state, and image source type (PAL or NTSC). In the situation in which the slave ATV-300's video output mode is NTSC, the video stream arriving from the AVA-300 is in PAL format, and the master ATV-300 is in PAL mode, the stream is too large for the ATV-300 to display and an error message is displayed.

The `slave mode Main Menu` is a restricted version of the `master mode` menu, since it is not possible for you to alter the Q-factor, frame rate, interlace state, or input channel of the incoming streams. Options are provided, however, to adjust the volume level of the audio output, save the current stream settings and configuration, change AVA-300s, display or remove the text banner, and exit `single-stream mode`.

9.8.5 Installing an ECM to Control a Remote AVA-300

To install an ECM to control a remote AVA-300, do the following:

1. Plug the ECM into the AUX DIN Connector on the rear panel of the ATV-300.
2. Configure the remote AVA-300 (which does not need an ECM) using the `Single-stream mode` menu option.
3. Save your configuration using the `Save Settings` menu option.
4. Power-up the ATV-300.

   The ATV-300 reads the stored configuration that was saved, resets the remote AVA-300, and then configures the remote AVA-300 with the settings stored in the ECM.

NOTE If the AVA-300 is power-cycled, the ATV-300 automatically reconfigures the AVA-300 when it comes back on-line.

NOTE You can only configure one video and one audio stream into a remote AVA-300 using this method. Also, as is the case with single stream mode, both a control (duplex), audio (simplex) and video (simplex) PVC must be open between the AVA-300 and ATV-300. If not, the ATV-300 assumes there is a problem and continually tries to reset the AVA-300.

CHAPTER 10 Troubleshooting

This chapter describes frequently-encountered setbacks and how to resolve them.

10.1 Can a single ATV-300 drive multiple monitors?

The ATV-300 is only able to drive a single S-Video and a single composite signal concurrently. Units which have multiple composite outputs should not be used to drive multiple monitors. A dedicated video amplifier must be used if multiple composite outputs are required.

10.2 Why do I get a poor-quality video signal from my ATV-300?

There are a number of factors you can check. First, look at your video source. If you are using a low-quality video tape as opposed to a laserdisk, overall video quality may suffer. Lower-quality video cameras can have the same affect. Connect your video source directly to your TV to gauge source quality.

Poor quality video cables and connectors can also degrade your video signal.

Finally, check your video stream settings. Q-factor/Variable Q-factor, bandwidth limitations, frame rate, and interlace-enabled can also affect your video signal.

10.3 Why does my ATV-300 audio distort?

Selecting `High` audio gain may cause loud audio signals to “clip” when they are sampled by the AVA-300, resulting in distortion of the audio reproduced at the ATV-300. A lower audio gain setting should be used when this happens.

Earlier versions of the AVA-300 may have different audio chips. If audio signals consistently sound distorted, it may be necessary for you to upgrade your AVA-300 firmware to the latest version. Also, AVA-300s with firmware versions earlier than 4.3 may exhibit consistent audio distortion.

10.4 Why do I get poor-quality workstation video/audio playback?

Different workstations and PCs are limited in their ability to process AVA video and audio. A common mistake is to overload the computer. Check the CPU usage on your machine if you suspect this is the case. Decompression video hardware can alleviate some of the processing overhead. Changing your stream settings may also be required.

10.5 What is an ATV-300 video plane?

The ATV-300 can display overlapping video streams. You can specify which video stream is “on top of” another by specifying a greater video plane number for the “top” (or frontmost) stream. This is commonly referred to as PIP (for picture-in-picture). The smaller picture may cover only 12% of the screen.

10.6 Is my AVA-200 compatible with the ATV-300?

AVA-200 audio is fully compatible. AVA-200 video is only compatible if the AVA-200 unit has the JPEG option fitted. Use the `avareset` program to check this. Also, AVA-200 JPEG video is not capable of transmitting interlaced streams which leads to a loss of vertical resolution.

10.7 Why am I having trouble entering Single-Stream Mode?

In the rare situation that the ATV-300’s default video mode is changed after saving the single-stream mode settings, the unit detects the change at the next reset or power-up and does not attempt to enter single-stream mode. In this case, however, the ATV-300’s stream configuration is still restored from the ECM.

10.8 Why am I losing video stream attributes in Master Single-Stream Mode when using an ECM?

In order to preserve the settings for brightness, contrast, color saturation, hue, and inversion, they must be written to an ECM using `Save Stream Settings`. The volume level and banner state are parameters local to the ATV-300 and are saved under `Save User Preferences`.

10.9 Why does my ATV-300 exit Single-Stream Mode?

While the ATV-300 is in master single-stream mode, it continuously monitors the streams arriving from the AVA-300. If the audio stream stops for longer than approximately 3 seconds, the ATV-300 assumes that the AVA-300 has been reset. It automatically attempts to re-contact the AVA-300, continuing to do so every 3 seconds until it is successful, or you change to another AVA-300, or it exits single-stream mode. A side-effect is that while the ATV-300 is attempting to make contact, the GUI may not respond for a period of about a second.

10.10 Why is my video signal displayed in monochrome from my ATV-300?

Your video jack may be plugged into an S-Video socket rather than a composite socket. Try experimenting with the video jack in different ATV-300 video connectors.

10.11 Why do I receive a blank or scrolling image when I power-up my ATV-300?

You may have a PAL/NTSC configuration error. Power cycle the ATV-300. Wait for 10 seconds. Press ❋ for PAL or # for NTSC on the Remote. The new configuration will be remembered automatically. If your image is blank at this point, check your cable connections.

10.12 How can I check that my AVA-300 is broken?

The easiest way to see whether or not your AVA-300 is broken is to connect it to a known, working ATV-300 and run the AVA-300 in `Single-stream mode`. If the ATV-300 cannot control the AVA-300, chances are that your AVA-300 is not functioning properly.

10.13 How can I check that my ATV-300 is broken?

The `Single-stream mode` test described above is a good indicator. Use a known, working AVA-300.

APPENDIX A Manual and Reference Pages

ATMDRIVERS(1) USER COMMANDS ATMDRIVERS(1)

NAME

atmdrivers - list ATM interfaces.

SYNOPSIS

atmdrivers

DESCRIPTION

The **atmdrivers** program shows the ATM interfaces on the current host that the SVA software is capable of working against. The output from **atmdrivers** lists each interface by name (which may be used as a value for the **-interface** argument which many SVA commands accept). Associated with each interface is a set of capabilities and any version and configuration information that may be helpful.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

atvconfig - configure a remote ATV unit.

SYNOPSIS

atvconfig -device <switch>:<port> | -vci <n>

[-vpi <n>]

[-vout [<opt1> <val1>] ... [<optN> <valN>]]

[-aout [<opt1> <val1>] ... [<optN> <valN>]]

[-ecm]

[-interface <str>]

DESCRIPTION

The **atvconfig** program allows a user on a remote workstation to load a video and audio stream configuration into an ATV unit. The user of this program must ensure that a control channel exists between the invoking workstation and the ATV. The **atvconfig** program will reset the ATV and so interrupt any current ATV operation.

The External Configuration Module (ECM) is an ATV hardware option which if activated will cause the ATV to load its configuration from the ECM on power up. The ATV will still respond to network control protocol messages in this state so it can be re-claimed or re-programmed by network control software if necessary. The **-ecm** option allows a user on a remote workstation to configure an ECM that is attached to an ATV unit.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–vpi <n>
Specify the control channel VPI.

–vout
Define a video stream to be loaded into the ATV. Any video stream parameters specified are interpreted by the **AtvVideoAttrParseArgs()** function. Later parameter specifications override earlier ones in the command line, so if the **source** keyword is used, it should come before any other keywords used to modify the default values for that **source**.

–aout
Define an audio stream to be loaded into the ATV. Any audio stream parameters specified are interpreted by the **AtvAudioAttrParseArgs()** function. Later parameter specifications override earlier ones in the command line, so if the **source** keyword is used, it should come before any other keywords used to modify the default values for that **source**.

–ecm
Load specified configuration into the ECM which is attached to the ATV unit.

–interface <str>
Specify ATM interface.

SEE ALSO

atmdrivers(1N), **spvc**(4N), **AtvVideoAttrParseArgs**(4N), **AtvAudioAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

atvdownload - upgrade ATV firmware.

SYNOPSIS

atvdownload -device <switch>:<port> | -vci <n>

[-vpi <n>]

[-f <filename>]

[-nochain]

[-noprompt]

[-keyfile <keyfilename> | -key <ab01cd02..>]

[-interface <str>]

DESCRIPTION

The **atvdownload** program allows a new version of firmware to be loaded into permanent storage on a remote ATV. The ATV firmware is distributed as a binary file which may be encrypted. The file format enables the **atvdownload** program to determine if the upgrade is encrypted. If it is, then a **decryption key** for the software release and the particular device must be available in order to perform the download. Decryption keys are typically distributed in key files. However, in special cases the -key option allows them to be specified directly.

If an upgrade file name is not specified then the **atvdownload** program will look for a suitable download file in the current directory. Such download files must be named as **atv300.version** where **version** is a number. If there are multiple candidates then the highest version number will be chosen. If the ATV is already loaded with the selected or a higher version number then **atvdownload** will query the operation. This query mode may be disabled using the -noprompt option.

By default the **atvdownload** program assumes that the remote ATV may be part of a device chain. The program attempts to determine the number and type of devices that are attached to the remote switch port. **atvdownload** will only perform an upgrade if the head device which is connected to the remote switch port is an ATV. Note that certain old firmware version ATV systems are not capable of responding to this device identification discovery process in which case **atvdownload** will fail over to using a legacy mode download operation. Alternatively, the -nochain option may be used directly.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–vpi <n>
Specify the control channel VPI.

–f <filename>
Specify the ATV binary upgrade file.

–nochain
The remote ATV is not capable of responding to device discovery requests.

–noprompt
Do not query user on download version discrepancies.

–keyfile <keyfilename>
Specify the file in which keys for encrypted downloads are to be searched for. A key file contains a set of decryption keys each to be used with a specific ATV device serial number and encrypted upgrade file.

–key <ab01cd02..>
Specify a decryption key directly rather than searching a key file.

–interface <str>
Specify the local ATM interface.

SEE ALSO

atmdrivers(1N), **spvc**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

atvmenu - display menus on an ATV and act on the user's selections.

SYNOPSIS

atvmenu -at <device>[@<trader-host>]

[-at_auth <password>]

[-title <"Menu Title">

[-item <"menu item">

[-menu | -exit]

[-<key>]... [<"shell command">]

] ...

]

[-menufile <filename>]

[-menufile -]

[-passwdfile <filename>]

[-trader <trader>1,<trader2>,...,<traderN>]

[-port <trader_port>]

DESCRIPTION

The **atvmenu** program causes a remote ATV device to display a menu of choices. When a user at the ATV selects one of these **–items,** the menu is removed and the corresponding action occurs. The options allow submenus to be created and navigated, and shell commands such as **svapatch(1N)** to be started and stopped.

SUBMENUS

Actions can generate or exit submenus by use of the **–menu** or **–exit** keywords.

An **–item** with the **–menu** keyword interprets the standard output of the given **shell command** (which must be present) as the **–title** and **–items** for a submenu to be pushed onto the menu stack.

An **–item** with the **–exit** keyword pops the submenu stack when it is selected. The submenu stack is also popped when the user presses the left-arrow key on the ATV infra-red remote controller.

A given **–item** may have either the **–menu** or the **–exit** keyword, but not both.

RESOURCES

An action can also execute an arbitrary **shell command** in a child process. The processes forked in this way are optionally labelled with user-specified **–<key>s.**

Each **–<key>** represents a resource which should only be held by one process at a time. For instance, a key called **–main** might represent the main video window and audio sink of an **svapatch** configuration.

Before it executes, an action sends the **SIGTERM** signal to each child process which holds one of the resources it requires, as specified by its **–<key>s.** Once the signalled processes have all exited, the action proceeds. If the action then forks a **shell command,** that process is recorded as the new holder of the given **–<key>s.** When it starts, **atvmenu –at device** acquires the session lock for the ATV **device.** The lock is released when **atvmenu** exits, which happens when either the user chooses an **–item** with the **–exit** key in the top-level menu or **atvmenu** itself receives the **SIGTERM** or **SIGINT** signal.

ENVIRONMENT

atvmenu exports various useful variables to the environment of its children. These are:

SVA_AT_DEV
the **device** given in the invocation of **atvmenu.**

SVA_AT_TRADER
any **trader** given in the invocation of **atvmenu.**

SVA_SESSION_AUTH_<device-identifier>
an authorization string for the session held by **atvmenu** at the target **device.**

The **svapatch(1N)** program uses these environment variables to allow it to run as a child of **atvmenu.** It shares the parent’s session at **device** rather than attempting to create its own. Also, the given **device** is used as the default sink device for patches with no explicit **–to** specification.

OPTIONS

–at device[@<trader-host>]
Set the name of the target ATV device. If **trader-host** is specified, the trader at that host will be searched for **device** before any other traders.

–passwdfile <filename>
Read passwords for password-protected managers from **filename.** If no **–passwdfile** option is given, the file **SVAauth** in the user’s home directory will be read by default, if it exists.

More than one **-passwdfile** option is allowed. The password for a given manager will be taken from the last **-passwdfile** which mentions that manger.

–at_auth <password>
This option can be used to override the password file when the target ATV device is password protected.

-title <**"Menu Title"**>
Specify the title of the current menu.

-item <"menu item">
Introduces a menu item in the current menu and specifies the text to appear for it on the ATV screen.

–menu
Specify that the current **–item** creates a submenu when selected.

–exit
Specify that the current **–item** leaves the current submenu when selected. If used in the top-level menu, specifies that the current **–item** quits **atvmenu.**

–<key>
Specify that the current **–item,** when selected, should acquire the resource labelled by **–<key>.** See the discussion in the section RESOURCES above.

<"shell command">
Specify that the current **–item,** when selected, should execute the given command in a child process once any necessary resources have been acquired. If this is a **–menu –item,** the output from the command is interpreted as the **–title** and **–items** for the submenu to be created.

–menufile <filename>
Interpret the contents of **filename** as the **–title** and **–items** of the top-level menu. If **filename** is **–,** read the standard input file.

–trader <trader1>,<trader2>,...,<traderN>
Monitor the trader(s) at the machine(s) **trader1,trader2,..., traderN** for the arrival and departure of device managers. If no trader options are supplied, the trader host defaults to the local machine.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

Note that the port number specified using this option applies only to the **atvmenu** program itself, not to any sub-processes (such as **svapatch**) that it may create. If you want a sub-process to use a non-default trader port, then you will need to specify **-port** for it as well.

EXAMPLES

```
atvmenu -at atv1 -title "Accept ?" \
 -item Yes -exit "echo yes" \
  -item No  -exit "echo no"

atvmenu -at atv1 -title "Channel Menu" \
  -item "BBC1"    -main "svapatch -from tuner bbc1-*" \
  -item "C4"      -main "svapatch -from tuner c4-*" \
  -item "Inset menu" -menu "cat inset.menu"
```

In the previous example, the file **inset.menu** might have the following contents:

```
-title "Inset Window Channel"
  -item "BBC1"
    -inset "svapatch -from tuner2 -video pip-bbc1 plane 1 x 64 y 32"
  -item "C4"
    -inset "svapatch -from tuner2 -video pip-c4 plane 1 x 64 y 32"
  -item "Back"
    -exit
```

Notice that by using the two keys **–main** and **–inset** above, the menus have independent control over the contents of the main video window and a smaller picture-in-picture window.

Also, none of the invocations of **svapatch** explicitly specifies a patch **–to atv1.** Instead, the sink device has been defaulted to the **SVA_AT_DEV** environment variable supplied by **atvmenu.**

SEE ALSO

svapatch(1N), **atvpatch**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is not supported on Windows 95.

NAME

atvpatch - display a menu on an ATV and perform specified patches

SYNOPSIS

atvpatch -at <device>[@<trader-host>] <"Menu Title">

[-passwordfile <filename>]

[-at_auth <password>]

[-item <"menu item text">

[-format [1up | 4up | pip | conf | conf3]]

[-from <device>[@<trader-host>]]

[-from_auth <password>]

[[-stream] [slot=]<source-pattern>]

[-video [slot=]<source-name> [[<opt1> <val1>] ...]]

[-audio <source-name> [[<opt1> <val1>] ...]]

[-specfile <filename>]

[-trader <trader1>,<trader2>,...,<traderN>]

[-port <trader_port>]

DESCRIPTION

The **atvpatch** program causes a remote ATV device to display a menu of choices. When a user at the ATV selects one of these, the menu is removed and a specified set of **svapatch-** style patches is set up.

After this, **atvpatch** continues to run, monitoring the streams and device managers involved just as **svapatch** does. In addition, **atvpatch** responds to subsequent menu requests from the ATV, deleting old connections and setting up new ones as necessary.

If it is interrupted **(SIGINT)** or terminated **(SIGTERM), atvpatch** attempts to shut down the connections it is maintaining and to delete the sinks it has created.

In order to display a menu and create sinks, **atvpatch** must acquire the session at the sink device manager. Existing sinks will be deleted at a sink device manager for which a session is successfully acquired.

OPTIONS

-at <device>[@<trader-host>] <"Menu Title">
Set the name of the target ATV device and the title of its menu. If **trader-host** is specified, the trader at that host will be searched for **device** before any other traders.

–passwdfile <filename>
Read passwords for password-protected managers from **filename.** If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists. The **-passwdfile** option must appear before the patch specifications to which it applies.

More than one **-passwdfile** option is allowed. The password for a given manager will be taken from the last **-passwdfile** which mentions that manager.

–at_auth <password>
This option can be used to override the password file when the target ATV device is password protected.

–item <"menu item text">
Introduce a new item which will appear as "menu item text" in the menu.

–from <device>[@<trader-host>]
Set the name of the current source device. This source device is assumed for each **–video** or **–audio** patch specification up to the next **–from** option. If **trader-host** is specified, the trader at that host will be searched for **device** before any other traders.

–from_auth <password>
This option can be used to override the password file when creating patches from password-protected (non-public) streams at the current source device to. It sets the current source stream password to **password.** This password is assumed for each **–video, –audio,** or **–stream** patch specification up to the next **–from_auth** or **–from** option.

-format [1up | 4up | pip | conf | conf3]
Specify a layout for the streams at the remote sink device. This option is interpreted in the same way as in **svapatch(1N)**.

[–stream] [slot=]<source-pattern>
Specify patches to the current sink device from all source streams at the current source device whose names match **source-pattern** under shell-style filename pattern matching (globbing). Note that since these patterns use the same wildcard characters as the Bourne shell, they may need to be quoted to prevent your shell from attempting to expand them itself.

The optional **slot=** syntax is used with the **-format** option as explained in **svapatch(1N)**.

–video [slot=]<source-name> [[opt1> <val1> ...]]
Specify a video patch to be created on selection of the current menu item. The patch is from the video source stream **source-name** at the current source device. This option is interpreted in the same way as in **svapatch,** except that the sink is always a remote device.

The optional **slot=** syntax is used with the **-format** option as explained in **svapatch(1N)**.

–audio <source-name> [[opt1> <val1> ...]]
Specify an audio patch to be created on selection of the current menu item. The patch is from the video source stream **source-name** at the current source device. This option is interpreted in the same way as in **svapatch,** except that the sink is always a remote device.

–specfile <filename>
Interpret the contents of **filename** as additional arguments to this **atvpatch.** If **filename** is **–,** use the standard input file.

–trader <trader1>,<trader2>,...,<traderN>
Monitor the trader(s) at the machine(s) **host1, host2, ..., hostN** for the arrival and departure of device managers. If no trader options are supplied, the trader host defaults to the local machine.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT,** then a default port of 2424 is used.

EXAMPLES

```
atvpatch -at tv1 "Channel Menu"  \

  -item "BBC1" -from bbc-ava "bbc1*"  \

  -item 'C4'   -from itv-ava "c4*"  \

  -item "Both" -format pip  \

     -from bbc-ava "bbc1*"
```

-from itv-ava c4-pip-jpeg

SEE ALSO

svapatch(1N), **atvmenu**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is not supported on Windows 95.

ATVRESET(1) USER COMMANDS ATVRESET(1)

NAME

atvreset - reset a remote ATV unit.

SYNOPSIS

atvreset -device <switch>:<port> | -vci <n>

[-vpi <n>]

[-ecm]

[-noversion]

[-interface <str>]

DESCRIPTION

The **atvreset** program enables a remote ATV to be reset. The user of this program must ensure that a control channel exists between the invoking workstation and the ATV.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–vpi <n>
Specify the control channel VPI.

–ecm
Load configuration from External Configuration Module (ECM) if present.

–noversion
Do not display the device version information which appears by default.

–interface <str>
Specify ATM interface.

SEE ALSO

atmdrivers(1N), **spvc**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

AVACONFIG(1) USER COMMANDS AVACONFIG(1)

NAME

avaconfig - configure AVA stream definitions.

SYNOPSIS

avaconfig -device <host>:<port> | -vci <n>

[-vpi <n>]

[-video [<opt1> <val1>] ... [<optN> <valN>]]

[-audio [<opt1> <val1>] ... [<optN> <valN>]]

[-serialin [<opt1> <val1>] ... [<optN> <valN>]]

[-serialout [<opt1> <val1>] ... [<optN> <valN>]]

[-pal | -ntsc]

[-interface <str>]

[-genlocked]

DESCRIPTION

The **avaconfig** program allows a user on a workstation to load video, audio and serial stream definitions into a remote AVA unit. The user of this program must ensure that a control channel exists between the invoking workstation and the AVA. The **avaconfig** program will reset the AVA and so interrupt any current AVA operation.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–vpi <n>
Specify the control channel VPI.

–video
Define a video stream to be loaded into the AVA. Any video stream parameters specified are interpreted by the AvaVideoAttrParseArgs() function. Later parameter specifications override earlier ones in the command line, so if the **source** keyword is used, it should come before any other keywords used to modify the default values for that **source**.

–audio
Define an audio stream to be loaded into the AVA. Any audio stream parameters specified are interpreted by the AvaAudioAttrParseArgs() function. Later parameter specifications override earlier ones in the command line, so if the **source** keyword is used, it should come before any other keywords used to modify the default values for that **source**.

–serialin
Define a serial input stream to be loaded into the AVA. That is serial characters will be received on the AVA serial port for transmission onto the ATM network. Any serial stream parameters specified are interpreted by the AvaSerialAttrParseArgs() function.

–serialout
Define a serial output stream to be loaded into the AVA. That is serial characters will be received from the ATM network for transmission on the AVA serial port. Any serial stream parameters specified are interpreted by the AvaSerialAttrParseArgs() function.

–pal
Configure the AVA for PAL video.

–ntsc
Configure the AVA for NTSC video.

–interface <str>
Specify ATM interface.

–genlocked
The AVA allows multiple analog video inputs to be encoded concurrently. The AVA has only one video analog to digital converter chip on board. In order that multiple video inputs be encoded concurrently the inputs must be gen-locked (synchronized). If this option is not set then **avaconfig** will reject such concurrent sampling requests.

EXAMPLES

avaconfig -device biggles.fore.com:1d3 -audio vci 45
avaconfig -vci 100 -video source atv vci 190

SEE ALSO

spvc(4N), **avareset**(1N), **ecmconfig**(1N), **atmdrivers**(1N), **AvaVideoAttrParseArgs**(4N), **AvaAudioAttrParseArgs**(4N), **AvaSerialAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

AVARESET(1) USER COMMANDS AVARESET(1)

NAME

avareset - reset a remote AVA unit.

SYNOPSIS

avareset -device <switch>:<port> | -vci <n>

[-vpi <n>]

[-ecm]

[-noversion]

[-interface <str>]

DESCRIPTION

The **avareset** program enables a remote AVA to be reset. The AVA is loaded with an empty video and audio schedule. The user of this program must ensure that a control channel exists between the invoking workstation and the AVA.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–vpi <n>
Specify the control channel VPI.

–ecm
Load configuration from External Configuration Module (ECM) if present and activated. Note that this option only refers to the video and audio configuration in the ECM. Activating additional configuration, such as ATM transmission interface parameters, requires the unit to be power cycled.

–noversion
Do not display the device version information which appears by default.

–interface <str>
Specify ATM interface.

SEE ALSO

atmdrivers(1N), **spvc**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

ecmconfig - program AVA External Configuration Module.

SYNOPSIS

ecmconfig -device <switch>:<port>| -vci <n>

-set | -clr

[-vpi <n>]

[-video [<opt1> <val1>] ... [<optN> <valN>]]

[-audio [<opt1> <val1>] ... [<optN> <valN>]]

[-serialin [<opt1> <val1>] ... [<optN> <valN>]]

[-serialout [<opt1> <val1>] ... [<optN> <valN>]]

[-ecmvpi <n>]

[-ecmvci <n>]

[-pal | -ntsc]

[-genlocked]

[-interface <str>]

DESCRIPTION

The External Configuration Module (ECM) is an AVA option which if activated will cause the AVA to load its configuration from the ECM upon power up. The AVA will still respond to network control protocol messages in this state so it can be re-claimed or re-programmed by network control software if necessary.

The **ecmconfig** program allows a user on a remote workstation to set and clear the video and audio configuration in an ECM which may be attached to an AVA unit. The user of this program must ensure that a control channel exists between the invoking workstation and the AVA. Changing an ECM value will reset the AVA and so interrupt any current AVA operation.

Loading and activating an ECM does not cause the specified configuration to be executed by the AVA immediately. In order for the AVA to load and execute the ECM configuration then the AVA must either be power-cycled or the **avareset** program with the **-ecm** argument may be used.

An ECM which has been configured on an AVA-200 system will not function when inserted into an AVA-300 system and vice-versa. Note that the ECM is still functional and can be simply re-configured using the **ecmconfig** program.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vci <n>
Specify the control channel VCI.

–set
Activate the attached ECM.

–clr
De-activate the attached ECM.

–vpi <n>
Specify the control channel VPI.

–video
When used in conjunction with the -set option will cause the AVA to load a video stream automatically on initialization. Any video stream parameters specified are interpreted by the AvaVideoAttrParseArgs() function.

–audio
When used in conjunction with the -set option will cause the AVA to load an audio stream automatically on initialization. Any audio stream parameters specified are interpreted by the AvaAudioAttrParseArgs() function.

–serialin
When used in conjunction with the -set option will cause the AVA to load a serial input stream automatically on initialization. Any serial stream parameters specified are interpreted by the AvaSerialAttrParseArgs() function.

–serialout
When used in conjunction with the -set option will cause the AVA to load a serial output stream automatically on initialization. Any serial stream parameters specified are interpreted by the AvaSerialAttrParseArgs() function.

–ecmvpi <n>
When used in conjunction with the -set option will cause the AVA to use VPI *n* for management PDUs after loading its configuration from the ECM. The default is VPI 0.

–ecmvci <n>
When used in conjunction with the -set option will cause the AVA to use VCI *n* for management PDUs after loading its configuration from the ECM. The default value is the VCI used for control at the device (see above).

–pal
When used in conjunction with the -set option will cause the AVA to configure for PAL video after loading its configuration from the ECM.

–ntsc
When used in conjunction with the -set option will cause the AVA to configure for NTSC video after loading its configuration from the ECM.

–genlocked
The AVA allows multiple analog video inputs to be encoded concurrently. The AVA has only one video analog to digital converter chip on board. In order that multiple video inputs be encoded concurrently the inputs must be gen-locked (synchronized). If this option is not set then **ecmconfig** will reject such concurrent sampling requests.

–interface <str>
Specify the local ATM interface.

EXAMPLE

ecmconfig -set -vci 100 -video source atv chan 3

SEE ALSO

spvc(4N), **avareset**(1N), **avaconfig**(1N), **atmdrivers**(1N), **AvaVideoAttrParseArgs**(4N), **AvaAudioAttrParseArgs**(4N), **AvaSerialAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

killmgr - tell managers to close down.

SYNOPSIS

killmgr <manager1>,<manager2>,..,<managerN> | -all

[-force]

[-trader <host1>,<host2>,..,<hostN>]

[-port <trader_port>]

[-passwdfile <filename>]

[-localhost <hostname>]

DESCRIPTION

The **killmgr** program allows a user on a workstation to cause the specified managers to close down and exit. The correct password is required when killing password protected managers. By default, if a user has a session lock at one of the managers specified, or if any streams are currently active, the kill will fail, but this behavior may be overridden by the **–force** option, which will cause the manager to exit even if a session was in progress or any streams active. Instead of specifying a list of managers, the user may specify the **–all** option, which causes **killmgr** to attempt close down every manager it finds in the specified traders.

OPTIONS

<manager>
Specify one or more managers which to kill. You may specify **<manager>@<host>** as a shorthand for **manager** and **-trader host**.

–all
Use of this option instead of a list of managers will cause **killmgr** to close down every manager it finds in the specified traders.

–force
Use of this option will cause all managers to close down, even those that have a session in progress or active streams.

–trader <host1>,<host2>,..,<hostN>
Specify one or more hosts which are running SVA trader processes. If no **–trader** argument is given then **killmgr** will use any trader hosts specified by the **NRLTRADERS** environment variable. If this environment variable is not specified then **killmgr** will default to assuming that there is a trader running on the same host as itself.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADER-PORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

–passwdfile <filename>
Read passwords for password-protected managers from **filename**. If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists. More than one **-passwdfile** option is allowed. The password for a given manager will be taken from the last **–passwdfile** which mentions that manager.

–localhost <hostname>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

OPTIONS

killmgr ava1 atv1 chain2@ariel
killmgr -force ava1
killmgr -all -trader ariel

SEE ALSO

svamgr(1N), **mgrls**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

listchain - list devices on chain.

SYNOPSIS

listchain -vci <n> | -device <switch>:<port>

[-vpi <n>]

DESCRIPTION

The **listchain** program lists the devices that are present on a device chain.

OPTIONS

–vci <n>
Specify the control channel VCI.

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page.

–vpi <n>
Specify the VPI to be used between the remote device(s) and the switch. Please note that the device(s) should be receiving signalling cells from only one signalling instance on the switch to which they are attached. For example, if VPI 1 is being used for UNI signalling then the UNI signalling engine for VPI 0 must be disabled at the switch. Note also that the control channel must be configured to use VPI 0 at the workstation end even though a non zero VPI is used at the device.

SEE ALSO

probepvc(1N), **spvc**(4N), **svamgr**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is not supported on Windows 95.

NAME

mgrls - list and describe SVA managers and their streams

SYNOPSIS

mgrls [-trader <host1>,<host2>,...,<hostN>]

[-localhost <hostname>]

[--port <n>]

[-m <manager>] [-s [<stream>]]

[-all]

[-l]

DESCRIPTION

In its simplest invocation, **mgrls** will list the SVA managers currently registered in the specified traders. Its behavior can be modified with switches to display more detailed information about each manager, and also to obtain a list of streams available from a manager, and the definitions of those streams.

OPTIONS

–trader <host1>,<host2>,...<hostN>
Specify one or more hosts which are running SVA Traders. If no **–trader** argument is given then **mgrls** will use any trader hosts specified by the **NRLTRADERS** environment variable. If this environment variable is not specified then **mgrls** will default to assuming that there is a trader running on the same host as itself.

–localhost <hostname>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

–port <n>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADER-PORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

–m <manager_name>
mgrls will interrogate the specified manager to obtain a full description of its capabilities. Multiple such arguments may be given. You may specify **manager@host** as a shorthand for **-m manager** and **-trader host**.

–s [<stream>]
The simple -s option (which may only be used in conjunction with a single **-m** option) causes **mgrls** to obtain not only the device description from the manager but also the list of audio, video and serial streams exported by that device. Streams that have sinks running against them will have their names prefixed with an asterisk.

–all
mgrls will interrogate **all** of the managers it finds registered in the specified list of traders. Use this option with caution as it may result in large amounts of network traffic. You may not specify this option in conjunction with any **-m** options.

–l
Display more information on managers. If this switch is on its own, this information includes the manager host, PID and UID. In conjunction with **-all**, this is extended to the complete device description.

Specifying the name of a stream after the -s switch causes mgrls to display the full definition for that stream.

If the trader list contains the names of non-existent hosts, **mgrls** will complain and exit. If the trader list contains the names of hosts that are not running trader processes, they will be ignored after a timeout period.

EXAMPLES

mgrls	(just manager names)
mgrls -l	(hosts, PIDs & UIDs)
mgrls -all	(manager info for all managers)
mgrls -l -all	(full manager info for all managers)
mgrls -m ava1	(manager info for ava1)
mgrls -m ava1 -l	(full manager info for ava1)
mgrls -m ava1 -s	(manager info and stream list for ava1)
mgrls -m ava1 -s mono	(ava1 manager info and mono stream defn)

SEE ALSO

SVA User's Manual, rtds-defaults(1N), **trader**(1N), **svaconfig**(1N).

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

nrlsh.

SYNOPSIS

nrlsh.

DESCRIPTION

nrlsh is the underlaying binary used by **mgrls** and **svaconfig**. It should not be invoked directly by users.

SEE ALSO

mgrls(1N), **svcconfig**(1N).

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NRLWISH(1) USER COMMANDS NRLWISH(1)

NAME

nrlwish.

SYNOPSIS

nrlwish.

DESCRIPTION

nrlwish is the underlaying binary used by **svc-rtds** and should not be invoked directly by users.

SEE ALSO

svc-rtds(1N).

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

probepvc **-** check for availability of PVC VPI/VCIs on the local workstation or PC.

SYNOPSIS

probepvc [-interface <str>]*

[-vpi <n>]*

[-vci <n>]+

DESCRIPTION

The **probepvc** program provides a simple means for testing for the availability of a given set of Permanent Virtual Circuit (PVC) VPI/VCIs on the local PC or workstation. In practice this means that no other application is using that VPI/VCI pair, it does not mean that the PVC is available or configured on the ATM network itself. The availability of multiple PVCs may be tested with a single invocation of **probepvc**. **probepvc** exits with a value of 1 if **any** of the requested PVCs are unavailable or a value of 0 if they are **all** available. Note that **probepvc** will exit as soon as it encounters an unavailable PVC.

The options are parsed from left to right and acted upon as they are read. The **–vci** argument results in an attempt to open a PVC with the specified VCI and currently specified ATM interface and VPI values as it is encountered. If multiple **–vci** arguments are given then each PVC will be tested as it is encountered on the command line. Note that **probepvc** will exit with a value of 1 as soon as it encounters an unavailable PVC and will not test any further PVCs specified on the command line. The **–interface** and **–vpi** options are used to specify the ATM interface and PVC VPI values respectively. Each value so specified then is then used for all subsequent PVC tests until it is overridden by another use of the same argument. If no interface or VPI values are specified on the command line then the default (usually the only) ATM interface is used along with a VPI of 0.

OPTIONS

–interface <str>
Specify an alternate ATM interface name (such as **xti:1** for example) for subsequent PVCs specified on the command line.

–vpi <n>
Specify a VPI to use for subsequent PVCs specified in the command line.

–vci <n>
Open a PVC with the specified vci and currently-in-force ATM interface and VPI.

EXAMPLES

probepvc -vci 45

Check the availability of a PVC with VPI 0, VCI 45 on the default ATM interface.

probepvc -vpi 1 -vci 23 -vpi 3 -vci 40 -vci 41

Check the availability of the 3 PVCs: the first VPI 1, VCI 23, the second VPI 3, VCI 40 and the third VPI 3, VCI 41.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

NAME

pserv - SVA patch server.

SYNOPSIS

pserv [-dir state_dir]

[-init]

[-users user_passwd_file]

[-no_auth]

[-serverport server_port]

[-snap_secs nsecs]

[-big_log_bytes nbytes]

[-trader host1,host2,...,hostN]

[-port trader_port]

[-cripple]

DESCRIPTION

The patch server, **pserv**, is a TCP server which implements the SVA Patch Server Protocol. Once a client program is connected and logged in to the server, it can issue instructions to control, connect and disconnect AVA and ATV devices by means of patch specifications in the style of **svapatch**(1N).

In order to use the patch server, a client program must log in with a user identity and password. The set of authorized users is stored in a file read by the patch server. By default, the file **SVAusers** in the current directory is used, but this can be changed with a command-line option.

The patch server provides facilities to retain information about the configurations and connections of AVAs and ATVs that have been established by its clients. This information is stored in a directory in the filing system and is **stable**: it persists independently of the client that created it, and survives client and server crashes.

The server's stable state is stored in the form of a checkpoint (or snapshot) of the complete state at some time, and a log of the operations performed since the last snapshot. It is possible to control how often the server considers taking a snapshot and how large a log must become before a snapshot is taken.

In order to allow for experimentation with the server without requiring real managed AVA and ATV devices to be available, the server can be **crippled**. A crippled server works just like a normal server except that all well-formed requests are treated as if they succeed immediately: no attempt is made to contact devices or traders.

OPTIONS

–dir state_dir
Read and write the stable state to and from files in the directory **state_dir**. If **state_dir** does not exist, create it. If no **–dir** option is given, the state directory defaults to the current directory.

–init
If this flag is present, and there is no existing stable state in the state directory, a fresh (empty) state is created. If the **–init** flag is NOT present, no fresh state will be created; if there is no existing stable state, **pserv** will report an error and exit.

–users user_passwd_file
Read the set of authorized users and their passwords from **user_passwd_file** instead of the default **SVAusers** file.

The format of this file is as follows: lines which consist only of whitespace, or whose first non-whitespace character is #, are treated as comments and ignored. Otherwise, each line gives the userid and password for a user separated by a single colon:

 # An example SVAusers file

user1:bPqw7rssT

user2:g4h0IIfdW

Care should be taken to set the permissions on the user password file appropriately: anyone who can read the file can log in to the patch server as one of the authorized users. Clients may also find it desirable to apply a one-way function such as **crypt(3C)** to passwords supplied by users before storing them in the user password file and using them in the patch server protocol's login challenge / response sequence. Doing this reduces the likelihood of another system being compromised if the patch server's password file is compromised and a user has been allowed to choose the same password value for both systems.

The patch server's security mechanisms are intended only to repel casual, rather than determined, cracking attempts.

–no_auth
Perform no authorization checking: any client may log in by quoting an arbitrary userid and password.

–serverport server_port
Listen for connection requests on the TCP port **server_port**. If this option is not present, the **server_port** defaults to decimal 2430.

Use of this option allows multiple patch servers to be run on the same machine. If the patch server fails to open the **server_port**, for example because another instance has already opened it, it reports that protocol initialization failed and exits.

–snap_secs nsecs
Check whether to take a stable state snapshot once every **nsecs** seconds. The default value is 30 seconds. A snapshot will be taken if the current log is both larger than the current snapshot and larger than a minimum size set with **–big_log_bytes**.

–big_log_bytes nbytes
Set the threshold log size above which a snapshot may be taken, provided the log is also larger than the current snapshot. The default value is 1024 bytes.

–trader host1,host2,...,hostN
Monitor the trader(s) at the machine(s) **host1, host2, ..., hostN** for the arrival and departure of device managers. If no trader options are supplied on the command line, the environment variable **NRLTRADERS** is used. If that is also empty, the trader host defaults to the local machine. Clients may also specify trader hosts in individual patch requests, as for **svapatch**(1N).

–port trader_port
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

–cripple
Cripple this patch server: no traders, AVA or ATV devices will be contacted and all well-formed requests will appear to succeed immediately.

EXAMPLES

pserv -dir master_state

pserv -dir test_state -init -cripple

SEE ALSO

psp(1N), **svapatch**(1N), **svamgr**(1N), Proposed PatchServer Specification SVAPSP / 1.0 v1.2.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is only supported in the Developer release.

This command is not supported on Windows 95.

NAME

psp - trivial SVA patch server protocol client.

SYNOPSIS

psp <host>[:<port>] [<userid> [<passwd>]]

DESCRIPTION

The **psp** program is a trivial command line client for the SVA patch server pserv (1N). Given a host and TCP port number to contact, it takes care of the details of connecting to the patch server on that host and port and logging in as a given user. Once connected, **psp** reads Tcl scripts from its standard input and transmits them in Patch Server Protocol requests to the server. Results and asynchronous messages from the server are in turn written to standard output.

If the standard input to **psp** is a terminal, a prompt giving the current request sequence number is printed when the next script is to be read. In interactive mode, results and messages from the server are printed in an easily-read format with one result or exception argument per line.

If the standard input to psp is NOT a terminal, the output is terser and easier for another program to parse. No prompts are printed, and each result or message is printed on exactly two lines. The first line gives the sequence number, response code and English response code string, separated by colons. The second line consists of a list of result or exception arguments, each enclosed in curly braces and concatenated:

15: 411: stream unknown

{video1}{ava1a}

For asynchronous messages, the sequence number (15 in the example above) is replaced by the string **STATUS**. If a request times out at the server, a single line containing the string **TIMEOUT** is printed. If no **<port>** is specified, the default value of decimal 2430 is used. If no **<userid>** is given, it defaults to the UNIX login name of the owner of the process. If no **<passwd>** is given, the **SVAauth** file in the caller's home directory is searched for an entry for **<host>**.

EXAMPLES

Interactive:

```
$ psp mymachine

1> session create {Test Session}

1: 200: ok

2> session list

2: 200: ok

  {session_1}

  {spqr1}

  {127.0.0.1}

  {Thu, 01 May 1997 12:19:19 +0100 (BST)}

  {Test Session}

  {Test Session}
```

Batch:

```
$ psp mymachine <<EOF

> session attach session_1 {Attached by test client}

> patch p1 -from ava -to atv -format 4up tl=video-jpeg-hf

> p1 show

> EOF

1: 200: ok


2: 200: ok
```

STATUS: 310: patch status

{session_1}{p1}{video-jpeg-hf}{ava}{4up:tl}{atv}{Lost}{source manager not found}

3: 510: lost patch

{session_1}{p1}{source manager not found}{Fresh}{Shown}{Lost}

SEE ALSO

pserv(1N), **svapatch**(1N), **svamgr**(1N), Proposed PatchServer Specification SVAPSP / 1.0 v1.2.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is only supported in the Developer release.

This command is not supported on Windows 95.

NAME

svachild - svamgr helper process.

SYNOPSIS

svachild [args]*

DESCRIPTION

The **svachild** program is invoked by the **svamgr** program as a helper process when configuring ATM device chains. There is no requirement for a user to invoke this program directly.

SEE ALSO

svamgr(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svaconfig - configure SVA manager streams.

SYNOPSIS

svaconfig -m <manager>[@<trader_host>] -s <stream>

[-localhost <hostname>]

[-passwdfile <filename>]

[-trader <host1>,<host2>,...,<hostN>]

[-port <trader_port>]

-video <key1> <val1> ... [<keyN> <valN>] |

-audio <key1> <val1> ... [<keyN> <valN>] |

-serial <key1> <val1> ... [<keyN> <valN>]

DESCRIPTION

The **svaconfig** program allows a user on a workstation to alter the parameters of a media stream definition in a managed SVA device. The user must specify a manager and a stream, and a set of key-value attribute pairs with which to override the stream's current settings. Legal attribute values are described below. If the syntax is incorrect, or the stream cannot be changed for any reason, **svaconfig** will return an explanatory message. In such failure modes, **svaconfig** will not change **any** of the stream parameters - it has an 'all-or-nothing' behavior.

OPTIONS

–m <manager>
Specify the managed device on which you wish to redefine a stream. You may specify **-m <manager>@<host>** as a shorthand for **-m <manager>** and **-trader <host>**.

–s <stream>
Specify the name of the stream whose definition you wish to alter.

–passwdfile <filename>
Read passwords for password-protected managers from **filename**. If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists.

More than one **-passwdfile** option is allowed. The password for a given manager will be taken from the last **–passwdfile** which mentions that manager.

–localhost <hostname>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

–trader <host1>,<host2>,...,<hostN>
Specify one or more hosts which are running SVA trader processes. If no **–trader** argument is given then **svaconfig** will use any trader hosts specified by the **NRLTRADERS** environment variable. If this environment variable is not specified then **svaconfig** will default to assuming that there is a trader running on the same host as itself.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

You may only use one of the following options at a time. Each marks the beginning of a list of attribute pairs which extends to the end of the command line, thus each must be the last option on the line.

–video <key1> <val1> ... [<keyN> <valN>]
Specify a list of key-value video attribute pairs with which to override the current stream definition. Legal values are shown below.

–audio <key1> <val1> ... [<keyN> <valN>]
Specify a list of key-value audio attribute pairs with which to override the current stream definition. Legal values are shown below.

–serial <key1> <val1> ... [<keyN> <valN>]
Specify a list of key-value serial attribute pairs with which to override the current stream definition. Legal values are shown below.

COMMON ATTRIBUTES

The following attributes are common to all streams and may be specified in conjunction with attributes specific to the type of the stream being changed.

public enable | disable
Set stream access to public (or private).

output enable | disable
Disable (or enable) source muting.

unit <number>
Specify the unit number (if on a chain).

qos ubr | vbr | cbr
Set the stream's Quality of Service parameter.

name <string>
Change the stream name.

vpi <number>
Set the VPI over which to send the stream. Note that setting this value to a non-zero value will disable SVCs and enable PVCs.

vci <number>
Set the VCI over which to send the stream. Note that setting this value to a non-zero value will disable SVCs and enable PVCs.

VIDEO ATTRIBUTES

There follows a list of video attributes, their legal values and a very brief explanation of what each refers to. Please see AvaVideoAttrParseArgs(4N) and the SVA manual for more information.

caption <string>
Specify the stream caption.

field either | interlace
Specify the field select.

svideo enable | disable
Set video inputs to S-Video (or Composite video).

qfactor <number> | variable
Set the JPEG Q-factor.

fps max | <number>
Set the frame rate.

limit <number>
Set the field size limit in bytes (variable Q-factor mode).

target <number>
Sets the target data rate in Mbps (variable Q-factor mode). This value, along with the framerate and the field select, is used to calculate a field size limit. You may not specify a target data rate and a field size limit. Note that the limit is **not** recalculated after changes in framerate and field select.

pcr <number>
Set the peak cell rate in cells/sec, or, if a decimal point is detected, in Mbps. The actual PCR may be altered slightly as the AVA is only capable of handling certain PCR values.

input <number> | 1a | 1b | 2a | 2b | 3a | 3b
Set the input number. (Called ‘channel’ in AvaVideoAttrParseArgs(4N)).

brightness <number>
Set the stream brightness control.

contrast <number>
Set the stream contrast control.

color <number>
Set the stream color control.

AUDIO ATTRIBUTES

These are the audio attributes. Please see AvaAudioAttrParseArgs(4N) and the SVA manual for more information.

gain <number>
Set the gain level.

format cd | dat | alaw | ulaw
Use one of four preset formats.

input <number>
Set the input number. (Called ‘channel’ in AvaAudioAttrParseArgs(4N)).

SERIAL ATTRIBUTES

These are the serial attributes. Please see AvaSerialAttrParseArgs(4N) and the SVA manual for more information.

delay <number>
Set the maximum buffering delay (latency) in milliseconds.

count <number>
Set the buffer size in bytes.

baud 300 | 600 | 1200 | 2400 | 4800 | 9600
Set the baud rate in bits per second.

EXAMPLES

svaconfig -m ava1 -s mono -video qfactor variable target 10 brightness -76
svaconfig -m ava1 -s mic -audio format dat gain 12
svaconfig -m ava1 -s serial -serial count 64 vci 127

SEE ALSO

svc-rtds(1N), **mgrls**(1N), **avaconfig**(1N), **AvaVideoAttrParseArgs**(4N), **AvaAudioAttrParse-Args**(4N), **AvaSerialAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svaman - find and display SVA manual pages

SYNOPSIS

svaman <name>

DESCRIPTION

svaman provides a convenient means of accessing the SVA manual pages. The program inspects the execution path and from this determines the installation directory of the SVA manual pages. Thus, if **svaman** is on your PATH variable there is no need to set your MANPATH variable in order to access the SVA manual pages.

SEE ALSO

svarun(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svamgr - manage a remote ATM device (chain).

SYNOPSIS

svamgr -device <switch>:<port> [-name <str>] |

-univci <n> -name <str> [-vci <n>] [-ilmivci <n>] |

-avamgr -vci <n> -univci <n> -ilmivci <n> -name <str>

[-uni30 | -uni31]

[-vpi <n>]

[-qos {cbr|vbr|show}]

[-pal | -ntsc]

[-genlocked]

[-configdir <dir>]

[-passwdfile <filename>]

[-trader <trader1>,<trader2>,..,<traderN>]

[-localhost <name>]

[-port <n>]

[-noexport]

[-contact <str>]

[-location <str>]

[-interface <str>]

DESCRIPTION

The **svamgr** program enables a remote ATM device or device chain to be controlled. This program presents a Remote Procedure Call (RPC) interface to clients that wish to share the device(s) being managed. The **svamgr** program translates these requests into primitive device requests.

Please note that it is strongly recommended that the SVA application **svarun** be used as a wrapper script when starting an SVA device manager.

The network administrator, when installing the service, must ensure that there exists a set of control PVCs between the workstation on which the **svamgr** program is to be run and the remote device (chain) or that the User Directed SPVC facility (recommended) is used.

The **svamgr** program registers its service interface with a **trader** service(s) on start up. The **trader** is similar to the SUN ONC **rpcbind** service. A single **trader** is able to handle the registration of multiple SVA manager interfaces. A client when wishing to discover what devices are available will first query the local **trader** services.

The **svamgr** program maintains a stream configuration database in a directory which may be specified by the **-configdir** option. The default directory to use is the current directory. Examples of the configuration file format may be found in the SVA software distribution. If no configuration file exists when **svamgr** starts up then it will look for a file called **svadefaults** from which to load its initial configuration.

OPTIONS

–device <host>:<port>
Specify the IP address and port number of the switch to which the device is attached. The use of this facility obviates the need to configure PVCs to the device; however, there are a number of caveats which are fully explained in the **spvc(4N)** manual page. Note if this option is used and no device name (see below) is specified then the device string will be used for the name, e.g., biggles.fore.com:1c3.

–name <str>
Specify the device name. This string will be used as the name of the device in the trader export record and also in the SNMP system group portion of the ILMI MIB. Note that ':' characters are not allowed in manager names and if present are replaced with '/'. Note that **–name** argument need not be specified if the **–device** argument is used, in which case the name of the manager will be the same as that specified for the **–device** argument.

–univci <n>
Specify the manager end VCI of the UNI PVC (duplex) between the **svamgr** program and the device(s). Note that the VCI at the device end of this PVC must be 37 if invoked in legacy mode otherwise the PVC is assumed to have the same VCI at each end of the circuit. Note that if the **svamgr** program is invoked in non-legacy mode and only the signalling VCI is specified then the ILMI and control VCIs are allocated by incrementing the signalling VCI.

–vci <n>
Specify the VCI of the control PVC (duplex) that is to be used by the **svamgr**. The PVC must use the same VCI at both the workstation and device ends.

–ilmivci <n>
Specify the manager end VCI of the ILMI PVC (duplex) between the **svamgr** program and the device(s). Note that the VCI at the device end of this PVC must be 48 if invoked in legacy mode otherwise the PVC is assumed to have the same VCI at each end of the circuit.

–avamgr
Force the **svamgr** to adopt the legacy **avamgr** mode of operation. Dynamic device discovery procedures will not be invoked. Please note that it is advised that AVA-300 systems are upgraded so that legacy mode operation is not required.

–uni30
The remote device (chain) will use the ATM Forum UNI 3.0 signalling protocol. This is the default.

–uni31
The remote device (chain) will use the ATM Forum UNI 3.1 signalling protocol.

–vpi <n>
Specify the VPI to be used between the remote device(s) and the switch for control and signalling channels. Please note that the device(s) should be receiving signalling cells from only one signalling instance on the switch to which they are attached. For example, if VPI 1 is being used for UNI signalling then the UNI signalling engine for VPI 0 must be disabled at the switch. Note also that the control and signalling channels must be configured to use VPI 0 at the workstation end even though they use a non-zero VPI at the device.

–qos {cbr|vbr|show}
This option causes the manager to use QoS (either CBR or VBR) profiles for the device control and signalling channels. The default, if this option is not used, is to use best effort (i.e. UBR) circuits. Note this option may only be used when the User Directed SPVC facility has also been selected. A series of default values are provided which should work for most networks consisting of high-speed (ATM25 and up) links. This option should only be used by people with a good understanding of ATM QoS. It is possible to override the default values using a series of environment variables, to view the full set of environment variables and the associated default values use **–qos show**.

–pal
Configure AVAs and ATVs for PAL video signals. Note that if no video format directive is specified then the **svamgr** will use the current time zone to derive the most likely default.

–ntsc
Configure AVAs and ATVs for NTSC video signals.

–genlocked
Some devices (e.g., AVA) allow multiple analog video inputs to be encoded concurrently. A device such as the AVA has only one video analog to digital converter chip on board. In order that multiple video inputs be encoded concurrently the inputs must be gen-locked (synchronized). If this option is not set then **svamgr** will reject such concurrent sampling requests.

–configdir <dir>
Specify the directory in which the **svamgr** configuration file should reside. This file has the same name as the manager with any white space characters replaced with underlines. The default directory is the current directory.

–passwdfile <filename>
Read passwords for password-protected managers from **filename.** If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists. More than one **–passwdfile** option is allowed. The password for a given manager will be taken from the last **–passwdfile** which mentions that manager. See **SVAauth**(4N) for more information on password files.

–trader <trader1>,<trader2>,..,<traderN>
Specify the host name of an additional **trader** or traders to which the **svamgr** should export registration information. Traders specified in this way will be used instead of the default traders specified in the **NRLTRADERS** environment variable, which should be formatted as a comma separated list of host names. Note that in addition to any traders specified by either of these two methods, the trader (if any) running on the local host will be always used as well.

–localhost <name>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

–port <n>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **-port** argument takes precedence over this. If no port has been specified using **-port** or **NRLTRADERPORT** then a default port of 2424 is used.

–noexport
Registration information for this **svamgr** is not be exchanged between peer **trader** processes.

–contact <str>
Specify the name of a contact to be used in the SNMP system group portion of the ILMI MIB.

–location <str>
Specify the location of the ATM device. This string will be used in the SNMP system group portion of the ILMI MIB.

–interface <str>
Specify ATM interface.

EXAMPLES

svamgr -device biggles.fore.com:1c3

svamgr -name pvc-mgr -univci 50 -ilmivci 60 -vci 70

svamgr -name consecutive-vcis -univci 160

SEE ALSO

spvc(4N), **svavideo**(4N), **AvaAudioAttrParseArgs**(4N), **AvaVideoAttrParseArgs**(4N), **AvaSerialAttrParseArgs**(4N), **svarun**(1N), **atmdrivers**(1N), **trader**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is not supported on Windows 95.

NAME

svapatch - configure and connect media sink devices to sources

SYNOPSIS

svapatch -from <device>[@<trader>]

-to <device1>[@<trader1>],...,<deviceN>[@<traderN>]]

[-stream] <source-pattern>

[-passwdfile <filename>]

[-from_auth <password>]

[-to_auth <password>]

[-format [1up | 4up | pip | conf | conf3]]

[[-stream] [slot=]<source-pattern>]

[-video [slot=]<source-name> [[<opt1> <val1>] ...]]

[-audio <source-name> [[<opt1> <val1>] ...]]

[-serial <source-name> [[<opt1> <val1>] ...]]

[-specfile <filename>]

[-trader <trader1>,<trader2>,...,<traderN>]

[-port <trader_port>]

svapatch -from device[@<trader-host>]

-to -local

[-stream] <source-pattern>

[-passwdfile <filename>]

[-from_auth <password>]

[-to_auth <password>]

[-format [1up | 4up | pip | conf | conf3]]

[[-stream] [slot=]<source-pattern>]

[-video [slot=]<source-name> [[<opt1> <val1>] ...]]

[-audio <source-name> [[<opt1> <val1>] ...]]

[-serial <source-name> [[<opt1> <val1>] ...]]

[-specfile <filename>]

[-trader <trader1>,<trader2>,...,<traderN>]

[-port <trader_port>]

svapatch -show_local_sinks

DESCRIPTION

The **svapatch** program allows a user on a remote workstation to connect video, audio and serial-line source streams from devices such as AVAs to sink devices such as ATVs, or to sinks on the local workstation. Sinks are created at the sink devices as appropriate. The sinks are configured by combining source properties with sink options specified by the user. SVC connections are created once sources and sinks are available.

The **svapatch** program continues to run after creating the initial sinks and SVC connections, monitoring the streams and device managers involved.

Whenever a source definition changes, for instance as a result of editing via **svc-rtds,** the corresponding sink configurations are recalculated and the sink device managers are informed.

Information about the synchronization of sources at a device is relayed to the sink device managers to which they are patched.

Finally, **svapatch** will attempt to maintain the specified connections in the face of device manager crashes, stream deletions and re-creations, connection failures and other error conditions.

If it is interrupted **(SIGINT)** or terminated **(SIGTERM), svapatch** attempts to shut down the connections it is maintaining and to delete the sinks it has created.

In order to create sinks, **svapatch** must acquire sessions at each of the sink device managers. Existing sinks will be deleted at sink device managers for which a session is successfully acquired. Passwords for password-protected managers are acquired by default from the user's SVA password file.

OPTIONS

–from <device>
Set the name of the current source device. This source device is assumed for each **–video, –audio, –serial,** or **–stream** patch specification up to the next **–from** option.

–to <device1>,<device2>,...,<deviceN>
Set the name(s) of the current sink device(s). These sink devices are assumed for each **–video, –audio, –serial,** or **–stream** patch specification up to the next **–to** option. If more than one sink device is supplied, a single patch specification giving a source stream name will cause multiple patches to be set up: one for each sink device.

To distinguish between different devices registered with separate traders under the same name, use the form **<device>@<trader-host>.** With this form, the trader at **trader-host** will be searched for **device** before any other traders specified with the **–trader** option. In the absence of such explicit instructions, whenever a manager called **device** is registered with any of the traders being searched, **svapatch** will attempt to move its patches to use the new **device**.

–to –local
Set the current sink device to the local workstation. This sink device is assumed for each **–video, –audio,** or **–stream** patch specification up to the next **–to** option. Sinks on the local workstation can display incoming streams; in addition, they can be used to record the data in a variety of formats and to a variety of destinations, including via pipes or shared memory. This is useful when connecting AVA sources to, for instance, MBone tools such as **vic** and **vat.** The particular formats, destinations and sink configuration options are specified as part of each **–video** or **–audio** patch specification.

Local sinks cannot at present be used for **–serial** sources.

–passwdfile <filename>
Read passwords for password-protected managers from **filename.** If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists. The **–passwdfile** option must appear before the patch specifications to which it applies.

–from_auth <password>
This option can be used to override the password file when creating patches from password-protected (non-public) streams at the current source device. This is useful if the current source device was started with a different password to that in the current password file (e.g. if it was started using a different password file). It sets the current source stream password to **password.** This password is assumed for each **–video, –audio,** or **–stream** patch specification up to the next **–from_auth** or **–from** option.

–to_auth password
This option can be used to override the password file when creating patches to a password-protected sink device. It sets the current sink device password to **password.** This is useful if the current sink device was started with a different password to that in the current password file (e.g. if it was started using a different password file). It sets the current source stream password to **password.** This password is assumed for each **–video, –audio,** or **–stream** patch specification up to the next **–to_auth** or **–to** option.

[–stream] [slot=]<source-pattern>
Specify patches to the current sink device(s) from all source streams at the current source device whose names match **source-pattern** under shell-style filename pattern matching (globbing). Note that since these patterns use the same wildcard characters as the Bourne shell, they may need to be quoted to prevent your shell from attempting to expand them itself.

The optional **slot=** syntax is used with the **–format** option explained below.

–video [slot=]<source-name> [[<opt1><val1>] ...]
Specify a patch to the current sink device(s) from the video source stream **source-name** at the current source device, providing additional options for the video sink.

The optional **slot=** syntax is used with the **–format** option explained below.

For a non **–local** sink device, the **optN** options are video stream parameters to be interpreted by the **AtvVideoAttrParseArgs()** function. The actual video stream definition at the source device will override all the given parameters except for **output, codec, channel, plane, x, y, scalex,** and **scaley.** The only scaling (other than one-to-one) currently supported by the ATV 300 is **scaley 50.**

For a **–local** sink device, the **optN** options are interpreted as an optional sink type name followed by a set of sink configuration commands. Use the **–show_local_sinks** option to find out what sink types are supported. If any of the options begins with a **–,** the whole set should be enclosed in **–args..–endargs;** for example:

"-video small-jpeg -args jpeg_null -record_start NamedShm Raw /tmp/vic_named -endargs".

The **vicsink** script is a convenient way of specifying a local video patch for the "ava" grabber built in to the **vic** MBone video conferencing tool.

–audio <source-name> [[<opt1> <val1>] ...]
Specify a patch to the current sink device(s) from the audio source stream **source-name** at the current source device, providing additional options for the audio sink. For a non **–local** sink device, the **optN** options are audio stream parameters to be interpreted by the **AtvAudioAttrParseArgs()** function. The actual audio stream definition at the source device will override all the given parameters except for **output, lgain, rgain, gain, codec, channel,** and **playout.**

For a **–local** sink device, the **optN** options are interpreted as an optional sink type name followed by a set of sink configuration commands. Use the **–show_local_sinks** option to find out what sink types are supported.

If any of the options begins with a **–,** the whole set should be enclosed in **–args..–endargs;** for example:

"-audio ulaw -args audio_null -record_start NamedRaw Raw /tmp/vat_named -endargs".

The **vatsink** script is a convenient way of specifying a local audio patch for a named pipe to be used by the **vat** MBone audio conferencing tool.

-format [1up | 4up | pip | conf | conf3]
Specify a layout for the streams at the remote sink device.

This option is a shorthand which specifies appropriate positions on an ATV screen for the video streams in a patch. A format consists of a number of video windows and an audio sink. Each video stream supplied is assigned to the video window in the format which it best fits; the measure of goodness-of-fit is the fraction of a window's area which would be covered by rendering the stream in that window. It is also possible to specify explicitly which stream is to be assigned to a given window; this will be explained below.

The first audio stream to be bound is assigned to the audio sink, so it generally makes sense to supply at most one.

The **1up** format gives the entire screen to one video stream. The **4up** format divides the screen in half both horizontally and vertically and expects up to four video streams. The **pip** format gives picture-in-picture: one window for the main picture, and a second small window inset in the main one. The **conf** format divides the screen into a large, roughly square window occupying two-thirds of the screen, and three smaller windows arranged in a vertical column to the right. The **conf3** format is similar, but there are only two windows in the vertical column.

If a **-format** option is in effect when a **-stream** or **-video** source is specified, the syntax **slot=source-name** can be used to assign **source-name** to the given **slot** in the current **-format**.

Note that in this case, the -format option must precede the source specification on the command line.

The standard manager configuration files **svadefaults** and **svadefaults.conf** contain video stream definitions suitable for use with the various **–formats.** The streams used with each format are as follows:

1up
slot **main**: jpeg-interlace (svadefaults)

4up
slots **tl**,**tr**,**bl**,**br**: video-jpeg-hf (svadefaults)

pip
slot **main**: jpeg-interlace; slot **small**: video-jpeg-pip (svadefaults)

conf
slot **main**: conference-jpeg; slots **top**, **middle**, **bottom**: third-jpeg (svadefaults.conf)

conf3
slot **main**: conf3-main; slots **top**, **bottom**: conf3-small (svadefaults.conf)

Notice that the **–format** option DOES NOT attempt to change the definitions of source streams to fit the specified format. If it is given the name of a video stream which is too large to fit into any unassigned window in the specified format, an error message will be printed when that fact is discovered.

Similarly, if more video streams are supplied than there are large enough slots in the specified format, an error message will be printed when that is discovered.

–serial <source-name> [[<opt1> <val1>] ...]
Specify a patch to the current sink device(s) from the serial-line source stream **source-name** at the current source device. The only meaningful option at present is the unit number at the sink device. The sink device cannot at present be **–local.** Note that a single **–serial** patch only creates a simplex connection from the source to the sink device. To create a duplex connection, specify one patch each way.

–specfile <filename>
Interpret the contents of **filename** as additional arguments to this **svapatch.** If **filename** is **–,** use the standard input file.

More than one **-passwdfile** option is allowed. The password for a given manger will be taken from the last **-passwdfile** which mentions that manager.

–trader <trader1>,<trader2>,...,<traderN>
Monitor the trader(s) at the machine(s) **<trader>,<trader2>,...,<traderN>** for the arrival and departure of device managers. If no trader options are supplied, the trader host defaults to the local machine.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

–show_local_sinks
Write the names of the supported local sink types to the standard error output, then exit. If other options are specified they will have no effect since **svapatch** will exit after displaying the local sinks.

ENVIRONMENT

svapatch reads various default values from environment variables. These allow **svapatch** to run as a child of another process while sharing its parent's sessions at device managers.

The values of the variables **SVA_AT_DEV** and **SVA_AT_TRADER,** if present, are used as the defaults for the **–to** option.

If a variable of the form **SVA_SESSION_AUTH_<device-identifier>** is supplied (by **atvmenu**) its value is taken as the authorization string for the session held by **atvmenu** at the identified device. In this case, **svapatch** shares the existing session at this device rather than attempting to create its own.

EXAMPLES

svapatch -from tuner -to tv1 -video bbc1-jpeg -audio bbc1-cd

If there are appropriately-named source streams at "tuner", the following command has the same effect as the previous one:

svapatch -from tuner -to tv1 'bbc1*'

**svapatch -to tv1@persil **

**-from vcr -video tape-jpeg plane 0 -audio tape-ulaw **

-from cameras -video cam1-jpeg plane 1 x 128 y 64

**svapatch -to tv1@persil,labtv@u.hypothetical.edu -format pip **

**-from vcr -video main=tape-jpeg -audio tape-ulaw **

-from cameras -video small=video-jpeg-pip

svapatch -from lecture-ava ‘vicsink vic‘ ‘vatsink vat‘

**svapatch -from ava1 -to ava2 serial **

-from ava2 -to ava1 serial

SEE ALSO

atvmenu(1N), **svc-rtds**(1N), **vicsink**(1N), **vatsink**(1N), **vic**(1), **vat**(1), **AtvVideoAttrParse-Args**(4N), **AtvAudioAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svarun - wrapper script for running AVA / ATV managers, traders and other SVA applications

SYNOPSIS

svarun binary [-configdir dir] [-log logfile] [args]*

DESCRIPTION

svarun provides a convenient means of running all SVA management processes, traders and other applications. It provides a number of features to simplify running these programs: automatic location of shared libraries, simple management of log files, checking for the availability of required network resources (ATM PVCs or UDP ports). **probepvc** is used to check for the availability of PVCs and **netstat(1)** is used for UDP ports.

The first argument to **svarun** determines which SVA application is to be run. The first argument must be the name of a valid executable, e.g. **svamgr**, **trader** or **netscape**. If the first argument is not an executable program (either a binary or a script) on the PATH, then **svarun** will default to running **svamgr**. In this way **svarun** can be used as a direct replacement for **svamgr**. All subsequent arguments are scanned by **svarun** and then passed through to the binary to be executed.

svarun provides an idempotent means of running managers - that is, running it multiple times in succession will have the same effect as running it once. This is particularly useful when writing scripts. **svarun** provides this functionality by examining its command line arguments for references to ATM PVCs or UDP ports and checking for their availability before running the specified binary. If the resources are not available then the binary will not be run. **svarun** takes care to avoid overwriting an existing log file when the specified binary cannot be run.

OPTIONS

–configdir dir
May be given for **svamgr** invocations (but not for other applications nor the trader) and specifies the directory in which the manager's configuration files are to be found and stored. If not given it will default to using the config directory in the SVA installation directory tree.

–log file
Argument specifies a log file to which all output from the manager is to be directed. The log file will not be truncated if **svarun** cannot run the requested manager or trader due to the non-availability of PVCs or UDP port. If a previous instance of the log file exists it will be renamed with the suffix **.old.**

[args]*
One or more arguments which are passed through to the binary to be executed.

EXAMPLES

svarun avaconfig -vci 156
Run **avaconfig** with the specified arguments.

svarun -ilmivci 155 -univci 156 -vci 157 -name AVA -configdir /tmp
Run a manager (svamgr) for an AVA using PVCs for control and signalling.

svarun -device biggles.fore.com:1c3 -name Chain -configdir /tmp
Run a manager for a chain of devices using SPVCs for control and signalling.

svarun svamgr -device biggles.fore.com:1c3 -name Chain -configdir /tmp
Same as above but specifying svamgr explicitly.

svarun trader -log /tmp/tlog
Run a trader.

SEE ALSO

svamgr(1N), **trader**(1N), **probepvc**(1N).

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svavers **-** show current SVA software version.

SYNOPSIS

svavers

DESCRIPTION

The **svavers** program displays the release number of the current SVA software release. It is useful to quote this when making a support enquiry.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svavideo - report default video standard.

SYNOPSIS

svavideo

DESCRIPTION

The **svavideo** program shows the user the default video standard to be used by SVA applications such as **avaconfig** and **svamgr.**

The SVA software attempts to derive the video standard to use by first inspecting the VSTANDARD environment variable. If this is set to either PAL or NTSC then that standard will be used as the default. If the environment variable is not set then the current time zone is used to derive the most likely default e.g. "EST" (Eastern Standard Time) would be used to derive a default video mode of NTSC.

SEE ALSO

avaconfig(4N), **svamgr**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

SVC-RTDS(1) USER COMMANDS SVC-RTDS(1)

NAME

svc-rtds - Real Time Display Software for SVA.

SYNOPSIS

svc-rtds [-trader <trader1>,<trader2>,..,<traderN>]

[-localhost <hostname>]

[-port <trader_port>]

[-passwdfile <filename>]

[-edit [-full]]

DESCRIPTION

svc-rtds provides a Graphical User Interface (GUI) for playing, editing and saving video and audio streams generated by either AVA-200 or AVA-300's on a UNIX workstation or a PC. The user interface is designed to allow a large number of AVA's and their video and audio streams to be accessed. Streams can be edited and saved to local disk storage.

The SVA User's Manual provides a full explanation of the **svc-rtds** program and should be read first. It contains a “Basic Setup'' chapter to help you bring up your SVA system as quickly as possible. This manual page is intended to provide a quick reference for the command line options and X11 style resources supported by **svc-rtds**. For platform specific information, e.g. video cards supported, please refer to the Release Notes for this software.

svc-rtds makes use of the RPC-based AVA and ATV device managers provided with SVA. The SVA User's Manual explains the system architecture and the role of device managers in detail.

OPTIONS

–trader <host1>,<host2>,..,<hostN>
Specify one or more hosts which are running SVA Traders. If no **–trader** arguments is given then **svc-rtds** will use any trader hosts specified by the **NRLTRADERS** environment variable. If this environment variable is not specified then **svc-rtds** will default to assuming that there is a trader running on the same host as itself.

–localhost <hostname>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

–port <trader_port>
Specify an alternate UDP port to use for communication with the traders. The **NRLTRADERPORT** environment variable can also be used to specify an alternate port for the traders. Note, that the **–port** argument takes precedence over this. If no port has been specified using **–port** or **NRLTRADERPORT** then a default port of 2424 is used.

–passwdfile <filename>
Read passwords for password-protected managers from **filename.** If no **–passwdfile** option is given, the file **SVAauth** in the user's home directory will be read by default, if it exists.

More than one **–passwdfile** option is allowed. The password for a given manager will be taken from the last **–passwdfile** which mentions that manager.

–edit
By default **svc-rtds** does not allow video and audio streams to be edited. Specifying this option enables the editing of video and audio streams.

–full
By default **–edit** prevents certain stream parameters from being edited, this is so as to simplify the user interface and restrict the potential for configuration mismatches. If this option is specified then the full set of stream parameters can be edited.

EXAMPLES

svc-rtds

A trader is running on the same host as **svc-rtds** and editing is not required.

svc-rtds -trader persil,ariel -edit
Traders are running on hosts persil and ariel; editing is required.

RESOURCES

Many of the default values used by **svc-rtds** can be configured using a similar mechanism to X11 resources. These values are specified in **rtds-def** files which are documented in the **rtds-def**(4N) manual page.

SEE ALSO

SVA User's Manual, **rtds-def**(4N), **trader** (1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

trader - remote service registration.

SYNOPSIS

trader [-trader <trader1>,<trader2>,..,<traderN>]

[-localhost <name>]

[-port <n>]

[-trace]

DESCRIPTION

The **trader** service is a network based Remote Procedure Call (RPC) service interface registration daemon. This service is analogous to the SUN ONC **rpcbind** facility. In order for an interface to be maintained in the service database it must be periodically refreshed. Any entry that is idle for 20 seconds is automatically deleted. The **trader** currently maintains all registrations in volatile store. Restarting the service will cause all current entries to be deleted. Note however that since most clients will be periodically refreshing their entries the old state should be quickly rebuilt. The **trader** service on a machine is accessible using a fixed UDP port (currently 2424).

It is possible to configure a **trader** with a list of other **trader** locations to which registration information should be passed. Note that registration information is only passed on if the client indicated that such a process is allowed in the original register RPC request. Note also that it is not possible to set up **trader** registration "chains". Information is only exchanged on a peer to peer basis.

OPTIONS

–trader <trader1>,<trader2>,..,<traderN>
Pass on RPC registration information to other **trader** services on the specified hosts. If this option is not used (or contains no resolvable names) then the **NRLTRADER** environment variable is inspected. The environment variable value should be formatted as a comma separated list of host domain names.

–localhost <name>
Directly specify the network interface to be used for RPC messages. This option is useful in the case of a multi-homed workstation for which routing between the two subnets is not allowed.

–port <n>
Specify UDP service port number. The default is 2424.

–trace
Enable registration and deletion tracing.

SEE ALSO

svamgr(1N), **svarun**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a9)

This command is not supported on Windows 95.

NAME

vatsink - print svapatch sink definition for vat

SYNOPSIS

vatsink <source-stream>

DESCRIPTION

The **vatsink** program prints on its standard output an **svapatch** local sink definition for a named pipe to be used by the **vat** MBone audio conferencing tool.

The argument is the name of the required stream at the source device. **vat** must be started up with a **–U /tmp/vat_named** option.

vatsink will typically be used as part of an **svapatch** command such as:
svapatch -from ava@svatrader ‘vatsink vat‘

SEE ALSO

svapatch(1N), **svc-rtds**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

vicsink - print svapatch sink definition for vic

SYNOPSIS

vicsink <source-stream>

DESCRIPTION

The **vicsink** program prints on its standard output an **svapatch** local sink definition for the named pipe and shared memory segment combination used by the "ava" grabber in the **vic** MBone audio conferencing tool.

The argument is the name of the required stream at the source device. **vicsink** will typically be used as part of an **svapatch** command such as:

svapatch -from ava@trader ‘vicsink vic‘

SEE ALSO

svapatch(1N), **svc-rtds**(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

AtvAudioAttrParseArgs - parse user command line audio arguments.

LIBRARY

Atv Library (libatv.a)

SYNOPSIS

#include <atv.h>

int AtvAudioAttrParseArgs(atv_audio_attr_t *attr,

int argc,

char **argv);

DESCRIPTION

The **AtvAudioAttrParseArgs()** function decodes an array of length *argc* of string pointers. The base of the array is referenced by *argv*. Sub sequences of this array are interpreted as options to apply to the audio attributes object *attr*.

vpi <n>

vci <n>

name <str>

mode <dat | cd | tel>

endian <big | little>

rate <n | dat | cd | tel>

playout <n>

latten <n>

ratten <n>

atten <n>

pack <n>

source <dat | cd | tel | atv>

Please note that if there is any string beginning with the character '-' in the argument list then parsing of the subsequent parameters is terminated.

The **AtvAudioAttrParseArgs()** function behaves in a manner analogous to its AVA equivalent except that the parameters refer to an audio output stream. The main differences from the AVA equivalent function are listed below.

The *playout* key is used to specify, in milliseconds, the length of the audio playout queue which the ATV unit will attempt to maintain for this stream. Samples will be inserted or deleted by the ATV as it attempts to "rate match" to the incoming audio stream.

The ATV is capable of attenuating the incoming audio stream prior to output (NB: the AVA is capable of both applying gain and attenuating its audio input). The *atten* key specifies the amount of attenuation to apply to the stream. Each unit specified is equivalent to 1.5 dB of attenuation. The *latten* and *ratten* keys may be used to independently specify the attenuation on the left and right channels respectively.

The *source* key is used to specify an audio stream definition that is compatible with the audio stream produced when the same source name is used with the **AvaAudioAttrParseArgs()** function.

Note that on return the audio attributes object has been mutated but not yet committed. No check has been made on the legality of the object produced. The function **AtvAudioAttrParse-Args()** will return the number of arguments consumed (possibly zero) on success and -1 if it has encountered an error parsing the options array.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

AtvVideoAttrParseArgs - parse user command line video arguments.

LIBRARY

Atv Library (libatv.a)

SYNOPSIS

#include <atv.h>

int AtvVideoAttrParseArgs(atv_video_attr_t *attr,

int argc,

char **argv);

DESCRIPTION

The **AtvVideoAttrParseArgs()** function decodes an array of length *argc* of string pointers. The base of the array is referenced by *argv*. Sub sequences of this array are interpreted as options to apply to the video attributes object *attr*. Note that if there is any string beginning with the character '-' in the argument list then parsing of the subsequent parameters is terminated.

vpi <n>

vci <n>

name <str>

qfactor <n> | variable

field <either | interlace>

reflect <enable | disable>

drop <n>

pack <n>

x <n>

y <n>

xtiles <n>

ytiles <n>

scalex <n>

scaley <n>

plane <n>

source <atv>

The **AtvVideoAttrParseArgs()** function behaves in a manner analogous to its AVA equivalent except that the parameters refer to an video output stream. The main differences from the AVA equivalent function are listed below.

The *reflect* key may be used to reverse the horizontal pixel ordering of a stream prior to display. This is useful in video-conferencing situations when a local source is viewed on the local display.

The *drop* key specifies the percentage of a frame that must be dropped before the ATV will not display the incoming image. If the decision is taken to drop a frame the last "complete" image that was received on the stream will be re-displayed.

The *plane* key is used to specify the overlay ordering of the incoming video streams. A video stream with a plane specification of one will be displayed on top of a video stream with a plane specification of zero. This function may be used to produce a picture-in-picture display.

The *source* key is used to specify an video stream definition that is compatible with the video stream produced when the same source name is used with the **AvaVideoAttrParseArgs()** function.

Note that on return the video attributes object has been mutated but not yet committed. No check has been made on the legality of the object produced. The function **AtvVideoAttrParseArgs()** will return the number of arguments consumed (possibly zero) on success and -1 if it has encountered an error parsing the options array.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

AvaAudioAttrParseArgs - parse user command line audio arguments.

LIBRARY

Ava Library (libava.a)

SYNOPSIS

#include <ava.h>

int AvaAudioAttrParseArgs(ava_audio_attr_t *attr,

int argc,

char **argv);

DESCRIPTION

The **AvaAudioAttrParseArgs()** function decodes an array of length *argc* of string pointers. The base of the array is referenced by *argv*. Sub sequences of this array are interpreted as options to apply to the audio attributes object *attr*. Note that if there is any string beginning with the character '-' in the argument list then parsing of the subsequent parameters is terminated.

vpi <n>

vci <n>

name <str>

mode <dat | cd | tel>

endian <big | little>

rate <n | dat | cd | tel>

qos <ubr | cbr | vbr>

chan <n>

level <mic | line>

gain <n>

mtu <n>

pack <n>

source <dat | cd | tel | atv>

The *vpi* and *vci* parameters specify the VPI and VCI that the audio stream will be transmitted on when the audio stream definition is loaded into the AVA. The default VPI is zero.

The *name* parameter is used to associate a text name with the audio stream. This name may then be used to reference the stream by higher level software. The default stream name is "unassigned".

The *mode* parameter is used to specify the format of the audio stream to be transmitted. The *endian* parameter may be used to force the endian-ness of the audio network format transmitted by the AVA to be either big or little endian.

The *rate* parameter is used to specify the sample rate of the audio stream generated by the AVA. If a numeric value is given the nearest legal sampling rate to the value specified will be selected. The discrete legal sample rate values are 5.5125, 6.615, 8.0, 9.6, 11.025, 16.0, 18.9, 22.05, 27.429, 32.0, 33.075, 37.8, 44.1 and 48.0 KHz.

The *qos* parameter specifies the type of Quality of Service (QoS) that should be specified for the ATM circuit used to transfer this stream. This parameter is interpreted by higher levels of the SVA software such as the **svamgr** program. An Unspecified Bit Rate (UBR) circuit makes no guarantees for the data being carried. A Constant Bit Rate (CBR) circuit allocates a fixed amount of constant bandwidth for the stream. A Variable Bit Rate (VBR) circuit is best defined if the stream is bursty.

The *chan* parameter specifies the audio input channel on the AVA from which the audio stream should be captured. Both the AVA-200 and AVA-300 each have three stereo audio input channels. Only one audio input may be selected at any point in time for transmission onto the ATM network. The default audio input is channel two on an AVA-200 and channel one on an AVA-300.

The *level* parameter specifies the signal level that is connected to the AVA audio input selected. The default is line level. Input one on both the AVA-200 and AVA-300 is capable of accepting a microphone level input. Due to the variety of different microphone types on the market it is recommended that pre-amplification on the microphone signal is applied prior to connection to the AVA at line level.

The *gain* parameter specifies the gain / attenuation to be applied to the audio signal. Each unit specified is equivalent to 1.5 dB of gain / attenuation. The default value for the *gain* parameter is 8 which leaves the signal level unchanged. The minimum *gain* value is zero which applies 12.0 dB of attenuation to the signal. The maximum *gain* value is 15 which applies 10.5 dB of gain to the signal.

An audio stream is transmitted as a sequence of AAL5 PDUs. The number of samples in each PDU is referred to as the packing factor. The audio packing factor may be specified directly by using the *pack* parameter. Alternatively, the *mtu* parameter may be used to specify the maximum size audio PDU that should be transmitted (the default is 4096 bytes) by the AVA. In this case the packing factor is calculated to be the largest legal value that does not violate the MTU constraint.

The *source* keyword provides a shorthand means of specifying a set of stream parameters. For example *source cd* specifies stereo, 16-bit audio sampled at 44.1 KHz.

Later parameter specifications override earlier ones in the command line, so if the **source** keyword is used, it should come before any other keywords used to modify the default values for that **source**.

Note that on return the audio attributes object has been mutated but not yet committed. No check has been made on the legality of the object produced. The function **AvaAudioAttrParseArgs()** will return the number of arguments consumed (possibly zero) on success and -1 if it has encountered an error parsing the options array.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

AvaSerialAttrParseArgs - parse user command line serial arguments

LIBRARY

Ava Library (libava.a)

SYNOPSIS

#include <ava.h>

int AvaSerialAttrParseArgs(ava_serial_attr_t *attr,

int argc,

char **argv);

DESCRIPTION

The **AvaSerialAttrParseArgs()** function decodes an array of length *argc* of string pointers. The base of the array is referenced by *argv*. Sub-sequences of this array are interpreted as options to apply to the serial attributes object *attr*. Note that if there is any string beginning with the character '-' in the argument list then parsing of the subsequent parameters is terminated.

vpi <n>

vci <n>

name <str>

baud <str>

delay <n> | max

count <n> | max

qos <ubr | cbr |vbr

The *vpi* and *vci* parameters specify the VPI and VCI that the serial stream circuit will use for communication on the ATM network. The default VPI is zero.

The *name* parameter is used to associate a text name with the serial stream. This name may then be used to reference the stream by higher level software. The default stream name is "unassigned".

An AVA-300 system is able to implement a variety of buffering strategies for serial input data prior to transmission on to the ATM network. The *delay* and *count* parameters are used to set the particular strategy required.

The *count* parameter is used to set the maximum number of characters that may be accumulated in the input buffer prior to transmission on to the network (currently the maximum size of this buffer is 32 characters).

The *delay* parameter is used to set the maximum amount of time (in milli-seconds) that a character can remain in the AVA input buffer prior to transmission on to the ATM network (currently the maximum delay that can be set on an AVA-300 system is 4000 milli-seconds).

The *qos* parameter specifies the type of Quality of Service (QoS) that should be specified for the ATM circuit used to transfer this stream. This parameter is interpreted by higher levels of the SVA software such as the **svamgr** program. An Unspecified Bit Rate (UBR) circuit makes no guarantees for the data being carried. A Constant Bit Rate (CBR) circuit allocates a fixed amount of constant bandwidth for the stream. A Variable Bit Rate (VBR) circuit is best defined if the stream is bursty.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

AvaVideoAttrParseArgs - parse user command line video arguments.

LIBRARY

Ava Library (libava.a)

SYNOPSIS

#include <ava.h>

int AvaVideoAttrParseArgs(ava_video_attr_t *attr,

int argc,

char **argv);

DESCRIPTION

The **AvaVideoAttrParseArgs()** function decodes an array of length *argc* of string pointers. The base of the array is referenced by *argv*. Sub-sequences of this array are interpreted as options to apply to the video attributes object *attr*. Note that if there is any string beginning with the character '-' in the argument list then parsing of the subsequent parameters is terminated.

vpi <n>

vci <n>

name <str>

mode <mono | rgb8 | rgb16 | rgb24 | rgb24-3 | jpeg>

endian <big | little>

field <either | interlace>

fps <n> | max

qos <ubr | cbr | vbr >

chan <n> | 1a | 2a | 3a | 1b | 2b | 3b

svideo <enable | disable>

brightness <n>

contrast <n>

color <n>

qfactor <n> | variable

pcr <n>

target <n>

mtu <n>

pack <n>

x <n>

y <n>

xtiles <n>

ytiles <n>

width <n>

height <n>

scalex <n>

scaley <n>

source <atv>

The *vpi* and *vci* parameters specify the VPI and VCI that the video stream will be transmitted on when the video stream definition is loaded into the AVA.

The *name* parameter is used to associate a text name with the video stream. This name may then be used to reference the stream by higher level software. The default stream name is "unassigned".

The *mode* parameter is used to specify the format of the video stream to be transmitted. Note that all of the video modes listed above produce big-endian format streams. The *endian* parameter may be used to force the endian-ness of the video network format transmitted by the AVA to be either big or little endian.

The *field* parameter specifies if the stream is to be interlaced or non-interlaced. If the stream is non-interlaced then the AVA scheduler is free to assign the stream to either the odd or even fields in the AVA transmit schedule.

The *fps* parameter is used to specify the number of frames per second for the video stream. If the video stream is configured to be interlaced then each frame transmitted will consist of two video fields. The maximum values for PAL and NTSC signals are 25 and 30 frames per second respectively. If a number of video streams are configured to be loaded into an AVA and the sum of their requested frame rates exceeds the maximum then the available schedule slots are shared out fairly between the video streams i.e. each stream will run at a frame rate lower than the one requested.

The *qos* parameter specifies the type of Quality of Service (QoS) that should be specified for the ATM circuit used to transfer this stream. This parameter is interpreted by higher levels of the SVA software such as the **svamgr** program. An Unspecified Bit Rate (UBR) circuit makes no guarantees for the data being carried. A Constant Bit Rate (CBR) circuit allocates a fixed amount of constant bandwidth for the stream. A Variable Bit Rate (VBR) circuit is best defined if the stream is bursty.

The *chan* parameter specifies the video input channel on the AVA from which the video stream should be captured. An AVA-200 has three video input channels whereas an AVA-300 system has six input channels. For an AVA-200 system the default video input is channel two composite. For an AVA-300 system the default video input is channel one (connector 1B) composite. The *svideo* parameter may be used to configure an input to sample an S-Video signal. In this case only three video inputs are available. S-Video channel one uses connectors 1A and 1B. S-Video channel two uses connectors 2A and 2B. S-Video channel three uses connectors 3A and 3B.

An AVA-300 unit is able to adjust the brightness, contrast and color of the video signal at source. The default setting for each of these values is zero. The *brightness*, *contrast* and *color* parameters may be used to adjust each of these settings at the AVA-300 respectively. The legal range for each parameter is from -500 to 500 inclusive.

The *x* and *y* parameters may be used to alter the top left co-ordinates of the video sample region. The co-ordinates are in pixel units. The default sampling co-ordinate is <0, 0>.

The *width* and *height* parameters specify the video sampling area in pixels (the default is the maximum area allowable). If the size of the video transmit region is not specified explicitly (see below) then the sample width and height will be rounded down to an integral number of tiles (i.e. a multiple of 8 pixels). In the case of a JPEG stream the width is rounded down to an integral multiple of 16 pixels (i.e. two tiles). After this transform has been applied the values are checked for legality. The maximum widths for PAL and NTSC images are 768 and 640 pixels respectively. The maximum heights are 288 and 240 pixels respectively.

The *scalex* and *scaley* parameters specify the video transmit region as a percentage of the sampling area (the default is 100%).

The *xtiles* and *ytiles* parameters provide an explicit mechanism to set the size of the video transmit region. The parameters are specified in tile units (i.e. blocks of 8 pixels). The maximum *xtiles* value for PAL and NTSC images are 96 and 80 tiles respectively. The maximum *ytiles* values are 36 and 30 tiles respectively. If present these parameters take precedence over any *scalex* and *scaley* parameters that have been specified.

The *pcr* parameter allows the peak cell rate in cells per second to be set for the associated video stream. If this parameter is entered as a floating point number then it is interpreted as the peak cell rate expressed in Mbps.

The *target* parameter allows the sustained target data rate for a variable Q-Factor stream to be specified. The target data rate is specified in Mbps. This figure is used to derive a target field size based on the frame rate defined for the stream. Please note that the target data rate must be less than the PCR. Setting the target data rate also configures the stream to be a variable Q-Factor JPEG stream as a side effect.

The *qfactor* parameter allows the Q-Factor for a JPEG video stream to be specified. Note that if variable Q-Factor is selected then the video stream will also be configured with a default target data rate.

Each video frame is transmitted as a number of AAL5 PDUs. The number of tiles in each AAL5 frame is referred to as the packing factor. The total number of tiles in the frame must be an integer multiple of the packing factor. A packing factor may be specified directly by using the *pack* parameter. Alternatively, the *mtu* parameter may be used to specify the maximum size PDU that should be transmitted (the default is 4096 bytes) by the AVA. In this case the packing factor is calculated to be the largest legal value that does not violate the MTU constraint. Note that for some video stream specifications no packing factor can be calculated and an error will be returned e.g. the packing factor for a JPEG stream must be a multiple of two.

The *source* key provides a mechanism whereby a single name can be used to specify a set of related stream attributes.

Note that on return the video attributes object has been mutated but not yet committed. No check has been made on the legality of the object produced. The function **AvaVideoAttrParseArgs()** will return the number of arguments consumed (possibly zero) on success and -1 if it has encountered an error parsing the options array.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

NrlAtmPvcOpen, NrlAtmPvcClose, NrlAtmSend, NrlAtmRecv, NrlAtmGetMtu, NrlAtmSetNonBlocking, NrlAtmSvcOpen, NrlAtmSvcBind, NrlAtmSvcListen, NrlAtmSvcConnect, NrlAtmSvcLocalAddr, NrlAtmSvcPeerAddr, NrlAtmSvcAccept, NrlAtmSvcClose, NrlAtmSvcCliOpen, NrlAtmSvcSvrOpen, NrlAtmEndpointEqual, NrlAtmStrToEndpoint, NrlAtmEndpointToStr, NrlAtmAddrTypeToStr, NrlAtmDataflowToStr, NrlAtmAalTypeToStr - ATM API (non standard).

Nrl Library (libnrl.a)

SYNOPSIS

#include <nrl.h>

```
typedef enum {
   Nrl_Atm_Aal0,
   Nrl_Atm_Aal1,
   Nrl_Atm_Aal2,
   Nrl_Atm_Aal3,
   Nrl_Atm_Aal4,
   Nrl_Atm_Aal5
} atmaaltype;

typedef enum {
   Nrl_Atm_Simplex,
   Nrl_Atm_Duplex,
   Nrl_Atm_Multicast
} atmdataflow;

typedef enum {
   TYPE_SPANS,
   TYPE_FORUM
} atmaddrtype;

int NrlAtmPvcOpen(char *device,
              unsigned int vpi_in,
              unsigned int vci_in,
              unsigned int vpi_out,
```

```
        unsigned int vci_out,
        atmaaltype aal);

int NrlAtmPvcClose(int fd);

int NrlAtmSend(int fd, char *buf, int nbytes);

int NrlAtmRecv(int fd, char *buf, int nbytes);

int NrlAtmGetMtu(int fd, unsigned int *mtu);

int NrlAtmSetNonBlocking(int fd);

int NrlAtmSvcOpen(char *device,
        atmaddrtype type,
        atmdataflow dataflow);

int NrlAtmSvcBind(int fd, atmendpoint *addr, int qlen);

int NrlAtmSvcListen(int fd,
         int *conn,
         atmendpoint *calling,
         atmqos *qos,
         atmaaltype *aal);

int NrlAtmSvcConnect(int fd,
          atmendpoint *dst,
          atmaaltype aal,
          atmqos *qos);

int NrlAtmSvcLocalAddr(int fd, atmendpoint *local);

int NrlAtmSvcPeerAddr(int fd, atmendpoint *peer);

int NrlAtmSvcAccept(int fd1, int fd2, int conn, int *newfd)

int NrlAtmSvcClose(int fd);
```

```
int NrlAtmSvcCliOpen(char *device,
              atmaddrtype type,
              atmdataflow dataflow,
              atmendpoint *local);

int NrlAtmSvcSvrOpen(char *device,
              atmaddrtype type,
              atmdataflow dataflow,
              atmendpoint *local,
              int qlen);

int NrlAtmEndpointEqual(atmendpoint *a,
                atmendpoint *b);

char *NrlAtmEndpointToStr(atmendpoint *endpt);

int NrlAtmStrToEndpoint(atmendpoint *endpt,
                char *str);

char *NrlAtmAddrTypeToStr(atmaddrtype addr);

char *NrlAtmDataflowToStr(atmdataflow dataflow);

char *NrlAtmAalTypeToStr(atmaaltype aal);
```

DESCRIPTION

The function **NrlAtmPvcOpen()** is used to allocate a handle representing an ATM PVC of the type requested by the parameter list. A return handle value of zero or greater indicates that the call has succeeded, otherwise -1 is returned indicating failure. On a UNIX platform the handle takes the form of a file descriptor which may subsequently be used as a parameter to the **select()** or **poll()** system calls. A *device* parameter of the null string indicates that the PVC is to be allocated on this particular host's default ATM interface. On hosts with multiple interfaces (possibly from multiple vendors) a non null string is used to uniquely identify each interface. See **atmdrivers** for a list of possible interfaces. The ATMDEVICE environment variable, if set, is used to override the default interface of the **NrlAtmPvcOpen()** call.

The function **NrlAtmPvcClose()** closes the PVC associated with handle *fd*. A return value of zero is used to indicate success, otherwise -1 is returned.

The function **NrlAtmSend()** sends *nbytes* of data starting from address *buf* on the ATM circuit referenced by *fd* encapsulated using the particular AAL that the connection was created with. The return value indicates the number of bytes sent.

The function **NrlAtmRecv()** is used to receive single SDUs from the circuit referenced by *fd*. The buffer to be used for the incoming SDU is described by the *buf* and *nbytes* parameters. The return value indicates the number of bytes that were present in the SDU received.

The function **NrlAtmGetMtu()** returns the maximum SDU that may be sent on the ATM connection referenced by *fd*. A return value of zero is used to indicate success and that *mtu* holds a valid value, otherwise a value -1 is returned.

The function **NrlAtmSetNonBlocking()** sets the connection referenced by *fd* into non-blocking mode. By default calls to **NrlAtmRecv()** will block in the operating system if there is no received SDU present when the call is made. If the connection has been set non-blocking and a call to **NrlAtmRecv()** is made when there is no data ready then the call will return immediately indicating that no data was available.

The function **NrlAtmSvcOpen()** is used to allocate a handle that may subsequently be used for SVC operation. The *type* parameter is used to specify what kind of signaling protocol is to be used. The *dataflow* parameter is used to specify the directional nature of the connection to be established with the returned handle. Refer to the **NrlAtmPvcOpen()** function for a description of the *device* parameter. A return value of -1 is used to indicate failure, otherwise a valid ATM SVC handle is returned.

The function **NrlAtmSvcBind()** allocates a local SAP to the embryonic SVC specified by *fd*. If the *addr* parameter is null then the next available SAP will be selected. If *qlen* is greater than zero then the lower layers are instructed that this is going to be a server SVC with *qlen* the maximum number of outstanding connect requests that can be queued locally.

The function **NrlAtmSvcListen()** is used to listen for a connect request from a remote ATM end point. On return *conn* contains a connection identifier to be used with the **NrlAtmSvcAccept()** function. On return *qos* specifies the peer's requested quality of service and *aal* the ATM adaption layer to be used for the connection. A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure.

The function **NrlAtmSvcAccept()** is used to accept an incoming connection request. The *fd1* parameter specifies the SVC handle that was used with **NrlAtmSvcListen().** The *fd2* parameter specifies the handle to use for the established connection. *fd1* and *fd2* can differ or be equal. *conn* must be a connection identifier returned by **NrlAtmSvcListen().** A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure. On some platforms (Win32/Winsock), a new SVC handle is returned in **newfd*, which must be used instead of *fd* for all subsequent calls (the original handle is closed by **NrlAtmSvcAccept()**). On all other platforms the original unchanged handle is returned in **newfd,* so in all cases using **newfd* will ensure correct operation.

The function **NrlAtmSvcConnect()** is used to establish a connection with a remote ATM end point. The *dst* field specifies the destination address for the connection. The *qos* and *aal* parameters indicate the requested quality of service and ATM adaption layer respectively. A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure.

The function **NrlAtmSvcLocalAddr()** returns the local end point address of the SVC referenced by *fd.* A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure.

The function **NrlAtmSvcPeerAddr()** returns the remote end point address of the SVC referenced by *fd.* A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure.

The function **NrlAtmSvcClose()** closes the SVC referenced by *fd.* A return value of zero or greater indicates that the call has succeeded, otherwise a value of -1 is returned indicating failure.

The function **NrlAtmSvcCliOpen()** is a convenience routine that calls both **NrlAtmSvcOpen()** and **NrlAtmSvcBind().** The *local* parameter must point to a valid area of store which will be used to return the local end point address allocated. A return value of -1 indicates failures, otherwise a valid SVC handle is returned.

The function **NrlAtmSvcSvrOpen()** is a convenience routine that calls both **NrlAtmSvcOpen()** and **NrlAtmSvcBind().** The *local* parameter must point to a valid area of store which will be used to return the local end point address allocated. A return value of -1 indicates failures, otherwise a valid SVC handle is returned.

The function **NrlAtmEndpointEqual()** returns a non zero value if the end points referenced by *a* and *b* are equal.

The function **NrlAtmStrToEndpoint()** converts from a string based representation of ATM addresses to the generic SVA ATM address structure. A return value of -1 indicates that the supplied ATM address string could not be parsed.

The function **NrlAtmEndpointToStr()** converts from the SVA ATM address structure to the string format described above. If the address structure describes an unknown address type then the string "unknown ATM address type" is returned.

The function **NrlAtmAddrTypeToStr()** returns a human readable string describing the given ATM *addr* type.

The function **NrlAtmDataflowToStr()** returns a human readable string describing the given ATM *dataflow* type.

The function **NrlAtmAalTypeToStr()** returns a human readable string describing the given ATM *aal* type.

SEE ALSO

atmdrivers(1N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

spvc - User Directed SPVC Facility.

DESCRIPTION

A large number of SVA utilities offer the choice of controlling the remote device either via Permanent Virtual Circuits (PVCs) or by specifying the switch IP address and port number to which it is attached. In the latter case the utility typically accepts syntax of the form **-device <switch>:<host>.**

In this case the SVA software attempts to utilize a facility known as User Directed Soft Permanent Virtual Circuits (SPVCs). This facility allows client applications, using a native ATM API, to establish an SVC to a particular switch port on the network without involving the remote attached device in any signalling interaction.

FORE Systems ForeRunner ATM switches offer a PNNI based SPVC facility which is pre-standards compliant. Additionally the Winsock2 and XTI native ATM API implementations available from FORE Systems have been extended to allow clients to act as sources of SPVC setup requests - hence the term User Directed SPVCs. Thus in order for the SVA software to utilize User Directed SPVCs it is necessary that the client be running ForeThought drivers of the appropriate version (ForeThought 5.x.x and later) and also that the target ForeRunner switch be running software of the appropriate version (ForeThought 4.1.x and later).

The target device is specified by quoting the IP address of the destination switch and the port to which the device is attached. The IP address may either be in dot notation form, e.g. 169.144.68.11, or may be the associated domain name. The SVA software uses this IP address to query, using SNMP, the NSAP prefix of the remote switch. Thus, there must be IP connectivity between the client machine and the target switch and also the client must be able to query the remote MIB. The switch port is specified using the standard FORE Systems notation, i.e. 1a1, 1a2, etc which is fully explained in the ForeRunner switch documentation if further information is required.

Note that it is possible to use non FORE Systems switches between the client and the remote ForeRunner switch to which the target device is attached so long as they transparently pass through the SPVC Information Element component of the connection establishment request.

It is FORE Systems intent to fully support evolving ATM Forum SPVC related standards. The pre-standards versions supported in this release of the SVA ForeThought software are made available as a value-add feature to our customers.

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

SVAauth - password file for SVA managers and clients

DESCRIPTION

The **SVAauth** file contains a list of manager name/password pairs. It is used by managers to set device passwords when starting up, and by client programs to supply a password when performing a restricted operation on a manager.

The format of the file is as follows. Each line is taken in turn. Leading whitespace is removed. If the line is empty or begins with a'#', it is treated as a comment and is ignored. The line is split into sections wherever one or more colons (':') or horizontal tab characters are encountered.

The first section becomes the manager name. The second section has leading and trailing spaces removed, and becomes the manager password. Any further sections are ignored.

Both manager names and passwords are restricted to the following characters: A-Z, a-z, 0-9, " " (space), "." (period), "_" (underscore), "-" (dash), "(" and ")" (brackets) and "/" (slash). A manager name may not consist only of digits. Manager names must be between 1 and 16 characters in length, and passwords must be between 6 and 16 characters long.

Note that the format described allows spaces in both manager names and passwords, and in-line comments. Here is an example **SVAauth** file:

```
#
#  (C) 1997 FORE Systems, Inc.
#
#  SVAauth (manager password file)
#


ava1:alaska    : Conference room
ava2:   utah


  # my office
  atv3:  new jersey: my desk
  ava3:  illinois: camera
```

office camera:maryland

end

By default this file is searched for in your home directory (on UNIX this is specified in **/etc/passwd**, on Windows this is the value of the **HOMEDRIVE/HOMEPATH** or **HOME** environment variables), but most SVA client programs allow you to specify an alternate location for this file using the **passwdfile** option.

SEE ALSO

svamgr(1N), **svc-rtds**(1N), **svapatch**(1N), **svaconfig**(1N), **killmgr**(1N), **atvmenu**(1N), **atvpatch**(1N), **ForeThought SVA-5.0 User's Manual**

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

NAME

svadefaults - default configuration of a managed ATM device

DESCRIPTION

When the **svamgr** program is executed, it attempts to read a configuration file which describes various parameters on the ATM device(s) being managed. The **svamgr** program searches for such a configuration in the following order:

1) A file corresponding to the -*name* parameter to **svamgr** is searched for

2) The **svadefaults** file is searched for

3) Defaults hard-coded into **svamgr** are used.

When **svamgr** exits, a file is written with the current configuration (which may have been altered during the execution of **svamgr** by SVA client software). The filename corresponds to the name of the device, overwriting the original file (in case 1 above) or creating a new file (in cases 2 and 3). The location of these files is controlled by the -*configdir* parameter to **svamgr** (see **svamgr**(1) for details)

Configuration files consist of *configuration object types* followed by curly-brace ("{}") delimited list of lines each containing a *configuration value*. Comments are lines beginning with a "%" character.

OBJECT TYPES

video | video.pal | video.ntsc

Video source stream definition. See **AvaVideoAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list. If the **video** key is used then the video stream definition is parsed/validated based on the video standard that the **svamgr** process is configured to use. If the **video.pal** key is used the stream will only be parsed/validated if the **svamgr** process is configured for PAL video. If the **video.ntsc** key is used the stream will only be parsed/validated if the **svamgr** process is configured for NTSC video. By using the latter two keys it is possible to define stream definitions in the configuration file with duplicate names only one of which will be selected based on the current video standard. The key **videoin** is equivalent to **video.**

videoout

Video sink stream definition. See **AtvVideoAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list.

audio

Audio source stream definition. See **AvaAudioAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list. The key **audioin** is equivalent to **audio.**

audioout

Audio sink stream definition. See **AvaAudioAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list.

serial

Serial Data (RS232 or equivalent) source stream definition. See **AvaSerialAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list. The key **serialin** is equivalent to **serial.**

serialout

Serial Data sink stream definition. See **AvaSerialAttrParseArgs**(4N) for a list of configuration values and their meanings. Streams are differentiated by their *name* value in the list.

properties

The manager stores certain values internally which control presentation on any managed ATV devices independent of the streams being displayed.These values are kept in the *properties* object. Property values are as follows: **captions on | off**

Enable / disable captions

caption-gravity <horizontal>-<vertical>

<horizontal> ::= (off | left | right | centre | edge)

"off" => centered in *window* "left" => at left of window "right" => at right of window "centre" => toward centre of *screen* "edge" => toward edge of *screen*

<vertical>::= (off | top | bottom | centre | edge)

"off" => centered in *window* "top" => at top of window "bottom" => at bottom of window "centre" => toward centre of *screen* "edge" => toward edge of *screen*

For example, **caption-gravity centre-top** with a 4-up display would result in the following positions for the captions in the 4 windows:

```
                              +-----+-----+
window: top-left captions at: top-right  |   1 | 2   |
     top-right          top-left   |     |     |
     bottom-left          top-right  +-----+-----+
     bottom-right          top-left  |   3 | 4   |
                              |     |     |
                              +-----------+
```

caption-font large | small

The *small* font is the normal ATV font which can be seen in any of the menus or error boxes. The *large* font is approximately twice the size of the *small* font.

date-format <format string>

The date display is specified using a subset of the format parameters to **strftime(3C):**

%%, %a, %A, %b, %B, %d, %e, %h, %j, %m, %n, %u, %U, %V, %w, %W, %x, %y, %Y, %Z

See **strftime(3C)** for details. **NOTE:** the implementation of **strftime** may differ across platforms. The user should ensure that the target platforms all support the format parameters used.

date-pos <anchor>=<position>

The position of the date display is specified by an anchor point relative to the display string and a position relative to the top-left of the display screen.

The anchor point is some combination of: (left | centre | right)-(top | centre | bottom)

The position is a (x,y) coordinate where (0,0) is the top-left corner of the display screen and (100,100) is the bottom right.

For example, to place the date centre-justified in the middle of the bottom of the screen, use *centre-bottom=50,100*

date-font large | small

See *caption-font* above

time-format <format string>

Similar to *date-format* with the following format parameters accepted:

%%, %H, %I, %k, %l, %M, %p, %S, %n, %Z

time-pos <anchor>=<position>

See *date-pos* above

time-font large | small

See *caption-font* above

In the stream definition (for both source and sink streams) if the VCI specified is zero (which is also the default) then the stream is defined to be an SVC template. For source streams this means that ATM endpoints may join using multicast SVC procedures. For sink SVC streams it is possible for clients to connect to the pre-defined stream using SVC connect procedures. If the VCI specified is non-zero then a PVC stream is created. For a source stream this means that the device, e.g. AVA-300, will start sending data out on the PVC selected. For a sink stream the device, e.g. ATV-300, will be configured to expect data of a certain type to be arriving on the specified VCI.

EXAMPLE

```
video
{
    name     "full-camera-2"
    caption  "Camera 2"
    fps      10
    field    interlace
    chan     2a
}

properties
{
    captions on
    caption-gravity centre-edge
    time-format "%H:%M:%S %Z"
    time-pos left-bottom=10,95
}
```


SEE ALSO

AvaAudioAttrParseArgs(4N), **AvaVideoAttrParseArgs**(4N), **AtvAudioAttrParseArgs**(4N), **AtvVideoAttrParseArgs**(4N), **AvaSerialAttrParseArgs**(4N)

RELEASE

FORE Systems Release: ForeThought 5.0 (SVA sva50a6)

APPENDIX B Environment Variables

The SVA software provides the `svaman` command as a wrapper script for the `man` command. You must set the `MANPATH` environment variable if you want to directly use `man` to access the SVA manual system.

To set the `MANPATH` environment variable, type the following:

```
prompt$ MANPATH=$MANPATH:$HOME/SVA-5.0.0/man; export MANPATH <Enter>
```

The `svarun` command is a wrapper script for invoking SVA applications. The `etc` directory must be added to your `PATH` environment variable.

To directly envoke SVA applications, type the following:

```
prompt$ PATH=$PATH:$HOME/SVA-5.0.0/etc; export PATH <Enter>
prompt% setenv PATH '$PATH':$HOME/SVA-5.0.0/etc <Enter>
```

Shared libraries are used by the SVA software on some operating system platforms in order to conserve space. To allow the software to locate the SVA shared libraries at run time, you may have to specify a search path (like `$PATH`).

On System V UNIX variants (such as Solaris 2.x and IRIX 5.x), the environment variable `LD_LIBRARY_PATH` is used for this purpose:

```
prompt$ LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/SVA-5.0.0/lib; export
LD_LIBRARY_PATH <Enter>
prompt% setenv LD_LIBRARY_PATH '$LD_LIBRARY_PATH':$HOME/SVA-5.0.0/lib
<Enter>
```

APPENDIX C PVC Control Channels

Many of the tools in the SVA software release require ATM circuits to be open to the remote device in order to function. The User-Directed SPVC facility makes this easy by allowing you to specify the remote device by naming the switch and the port number to which the remote device (or CellChain) is attached. The application then uses the User-Directed SPVC function to automatically create the required ATM circuit(s).

However, the User-Directed SPVC facility currently requires that your network be based primarily on FORE Systems equipment with a suitable software version.

NOTE

Refer to Section 4.4 if you are not sure if your network supports User-Directed SPVCs. If this facility is not available, then you must use the application in PVC control channel mode.

The on-line documentation and Appendix A detail how each command should be used in PVC mode. In order to illustrate the procedure, an example using the `avareset` program follows below.

NOTE

The following example assumes that you are using the AVA-300 configuration described in Chapter 4 (UNIX) or Chapter 5 (Windows). It is also assumed that the control workstation or PC is attached to port `1A2` on the same switch.

To use the `avareset` program, a bi-directional control PVC must be established between the AVA-300 and the workstation. Since the channel is bi-directional, you must configure two uni-directional channels on the switch (when using a *ForeRunner* ATM switch).

To configure the AVA-300 control channel on a *ForeRunner* ATM switch, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the local host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Configure the control channel in both directions, as in this example:

```
new 1A1 0 130 1A2 0 130

new 1A2 0 130 1A1 0 130
```

where 0 is the VPI and 130 is the VCI.

NOTE Do not allocate VCIs 0-31 since these are reserved by the ATM Forum and International Telecommunication Union (ITU).

NOTE If you are using a switch from another manufacturer, refer to its accompanying User's Manual to configure PVCs.

6. Once the control channel has been configured, use the `avareset` command to reset the remote AVA-300 as follows:

```
svarun avareset -vci 130
```

If the control channel is configured properly, the system displays version information similar to the following:

```
Hardware:300.6
Firmware:4.3
Serial No:96030045
Interface:ATM 155 Mbps
ECM:Yes
Release:Forethought 5.0.0 (SVA sva50a4)
```

If the control channel is not configured properly, the system displays the following error message:

```
avareset: failed to contact AVA ... retry
```

If you receive this error message, double-check your connections and/or your configuration syntax.

Other SVA device configuration programs operate similarly to `avareset`. AVA/ATV device managers require extra PVCs to be configured. These are detailed in Appendix E and Appendix F.

APPENDIX D ATV-300 Firmware Upgrade

The ATV-300 has on-board FLASH memory which is used to store the ATV-300 control program. The FLASH memory also stores the configurations for some of the re-programmable hardware devices on the ATV-300. You can use the `atvdownload` program over the ATM network to remotely load a new control program and hardware configuration into an ATV-300.

NOTE The following procedure assumes that the ATV-300 to be upgraded is attached to port `1B1` on switch `orion`, as described in the Basic Setup configuration.

NOTE If you are using a network configuration which does not support User-Directed SPVCs, (see Section 4.4) then refer to Appendix C.

To upgrade the ATV-300 firmware, do the following:

1. Obtain the new ATV-300 firmware either by contacting FORE Systems' Technical Support or receiving a new SVA software release. The file name reflects the version number; e.g., `atv300.18`. The ATV-300 firmware release is located in the top-level `firmware` directory of the SVA software distribution.
2. Go to the directory that contains the ATV-300 download file.
3. Terminate any manager process involving the ATV-300.
4. Make sure that you can correctly reset the ATV-300 by typing:

   `svarun atvreset -device orion:1B1`

   When successful, the ATV-300 version information is displayed. If not successful, please check your network configuration and also that no manager process is running against the ATV-300.
5. Download the new firmware to the ATV-300 by typing:

   `svarun atvdownload -device orion:1B1`

NOTE

The ATV-300 is reset by the `atvdownload` program and begins to execute with the new firmware configuration immediately.

6. Repeat step 4 to verify that the correct firmware version is now running in the ATV-300.

You have remotely upgraded your firmware using the `atvdownload` program.

NOTE

The `atvdownload` program's options are fully described in the `atvdownload` manual page located in Appendix A.

APPENDIX E svamgr: PVC Control Channels

This appendix describes how to manage AVA-300 and ATV-300 systems using PVC control channels as opposed to User-Directed SPVCs which are described in earlier chapters.

NOTE AVA-300s must have firmware version 4.3 or later in order for the configuration described in this appendix to work.

NOTE ATV-300s must have firmware version 300.14 or later in order for the configuration described in this appendix to work.

NOTE The examples worked through in this appendix are brief. They are intended to be used in conjunction with Chapter 4 and Chapter 5. These chapters describe in greater detail the interaction of device managers with the rest of the SVA software.

NOTE This appendix assumes that the AVA-300 and ATV-300 are attached to ports `1A1` and `1B1` respectively on the same *ForeRunner* switch. Also assumed is that the control workstation/PC is attached to port `1A2` on the same switch.

E.1 Setting Up the AVA-300 Manager

Each AVA-300 manager requires three bi-directional PVCs to be established between it and the remote device. These PVCs are referred to as the control, signalling, and ILMI PVCs.

To configure the PVCs on a *ForeRunner* ATM switch, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the local host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Configure the control PVC in both directions, as in this example:

   `new 1A1 0 130 1A2 0 130`

   `new 1A2 0 130 1A1 0 130`

6. Configure the signalling PVC in both directions, as in this example:

   `new 1A1 0 131 1A2 0 131`

   `new 1A2 0 131 1A1 0 131`

7. Configure the ILMI PVC in both directions, as in this example:

   `new 1A1 0 132 1A2 0 132`

   `new 1A2 0 132 1A1 0 132`

NOTE The above PVCs use example VCI values. The numbers allocated do not need to be consecutive, but this will simplify the manager command line invocation.

NOTE The PVCs must have the same value VCI at each end of the circuit.

NOTE Do not allocate VCIs 0-31 since these are reserved by the ATM Forum and International Telecommunication Union (ITU).

To start the AVA-300 manager, type the following at the workstation/PC which is to run the manager:

`svarun -name ava1 -vci 130 -univci 131 -ilmivci 132`

NOTE

Since consecutive VCIs have been allocated in this example setup, the following format is equivalent:

`svarun -name ava1 -univci 130`

NOTE

Please refer to the `svamgr` on-line manual page for more information on the options that are available when running SVA device managers.

The following message lets you know that the `manager` application is running:

`(Date and time): svamgr: started on host` *"<hostname>"*

In addition, a list of available video and audio streams follows the message.

You have started the AVA-300 manager and registered it with the trader.

E.2 Setting Up the ATV-300 Manager

Each ATV-300 manager requires three bi-directional PVCs to be established between it and the remote device. These PVCs are referred to as the control, signalling, and ILMI PVCs.

To configure the PVCs on a *ForeRunner* ATM switch, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the local host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Configure the control PVC in both directions, as in this example:

   `new 1B1 0 140 1A2 0 140`

   `new 1A2 0 140 1B1 0 140`

6. Configure the signalling PVC in both directions, as in this example:

   `new 1B1 0 141 1A2 0 141`

   `new 1A2 0 141 1B1 0 141`

7. Configure the ILMI PVC in both directions, as in this example:

   `new 1B1 0 142 1A2 0 142`

   `new 1A2 0 142 1B1 0 142`

NOTE

The above PVCs use example VCI values. The numbers allocated do not need to be consecutive, but this will simplify the manager command line invocation.

NOTE

The PVCs must have the same value VCI at each end of the circuit.

NOTE

Do not allocate VCIs 0-31 since these are reserved by the ATM Forum and International Telecommunication Union (ITU).

To start the ATV-300 manager, type the following at the workstation/PC which is to run the manager:

```
svarun -name atv1 -vci 140 -univci 141 -ilmivci 142
```

NOTE Since consecutive VCIs have been allocated in this example setup, the following format is equivalent:

```
svarun -name ava1 -univci 140
```

NOTE Please refer to the `svamgr` on-line manual page for more information on the options that are available when running SVA device managers.

The following message lets you know that the `manager` application is running:

```
(Date and time): svamgr: started on host "<hostname>"
```

You have started the ATV-300 manager and registered it with the trader.

APPENDIX F svamgr: Early AVA-300 Units

This appendix describes how to manage early AVA-300 firmware version units.

NOTE AVA-300s with firmware versions less than 4.3 must be managed using the process described in this appendix.

NOTE AVA-200s with firmware version 2.9 may be managed using the process described in this appendix.

NOTE The example worked through in this appendix is brief. It is intended to be used in conjunction with Chapter 4 and Chapter 5. These chapters describe in greater detail the interaction of device managers with the rest of the SVA software.

NOTE This appendix assumes that the AVA and control workstation/PC are attached to ports `1A1` and `1A2` respectively on the same *ForeRunner* switch.

F.1 Setting Up the AVA Manager

Each AVA manager requires three bi-directional PVCs to be established between it and the remote device. These PVCs are referred to as the control, signalling, and ILMI PVCs.

To configure the PVCs on a *ForeRunner* ATM switch, do the following:

1. Connect an ASCII terminal to the switch serial port or open a telnet session.
2. Login to the switch.
3. At the local host prompt, type **`config`** to get to the `configuration` submenu.
4. At the configuration prompt, type **`vcc`** to get to the `vcc` submenu.
5. Configure the control PVC in both directions, as in this example:

```
new 1A1 0 130 1A2 0 130

new 1A2 0 130 1A1 0 130
```

NOTE The PVCs must have the same value VCI at each end of the circuit.

NOTE Do not allocate VCIs 0-31 since these are reserved by the ATM Forum and International Telecommunication Union (ITU).

6. Configure the signalling PVC in both directions, as in this example:

```
new 1A1 0 37 1A2 0 131

new 1A2 0 131 1A1 0 37
```

NOTE The UNI signalling channel must be configured to VCI 37 at the AVA port.

7. Configure the ILMI PVC in both directions, as in this example:

```
new 1A1 0 48 1A2 0 132

new 1A2 0 132 1A1 0 48
```

NOTE

The ILMI signalling channel must be configured to VCI 48 at the AVA port.

To start the AVA manager, type the following at the workstation / PC which is to run the manager:

`svarun -avamgr -name ava1 -vci 130 -univci 131 -ilmivci 132`

NOTE

Please refer to the `svamgr` on-line manual page for more information on the options that are available when running SVA device managers.

The following message lets you know that the `manager` application is running:

`(Date and time): avamgr: started on host` *"<hostname>"*

In addition, a list of available video and audio streams follows the message.

You have started the AVA manager and registered it with the trader.

APPENDIX G Audio Conference Setup

This appendix provides an introduction to configuring an audio set-up for a high-quality bi-directional teleconference. While setting up a basic audio conference is fairly straightforward, achieving optimum results with the equipment and environment at your disposal usually requires some experimentation; it is impossible to test every possible setup and there are parameters which cannot be prescribed in this section, such as the exact models of audio components, the acoustics of the room, the background noise level, and the number and seating arrangements of participants. However, this section introduces basic concepts which can be applied to your choice of equipment and setting.

G.1 Hardware Requirements

You will need the following equipment at each of the videoconference sites to complete a bidirectional teleconference configuration:

- Sturdy table or desk to set the equipment on
- Heavy, shock-absorbent microphone stand
- CellChain consisting of one ATV-300 and one AVA-300)
- Color TV with video and stereo audio inputs.
- Microphone
- Video camera
- Mixer with microphone pre-amp
- Fiber optic cables
- Behringer Ultra-Curve DSP-8000 (or similar) graphic EQ unit with feedback eliminators (optional; required at *only one end* of the teleconference, for higher-quality audio)

  See Section G.4 for descriptions of parts and references.

G.2 Configuring the Hardware

This section details the procedure for setting up your audio equipment for a bidirectional teleconference. To set up your audio equipment, do the following:

1. Set up a CellChain at each site, connect it to the network and attach the TV and video camera.

NOTE

Refer to Chapter 4 or Chapter 5 for procedures that detail hardware connections, and Section 7.4 which discusses CellChains.

2. Run two `svapatch` programs to connect the two CellChains together, using the `cd audio` stream and the `jpeg-interlace` video stream, as in the following:

```
svapatch -from cellchain1 -to cellchain2 -video jpeg-interlace -audio cd

svapatch -from cellchain2 -to cellchain1 -video jpeg-interlace -audio cd
```

NOTE

Using S-video instead of composite video will give a better picture.

3. Place the TV on a table or desk, with the user facing it and seated comfortably about 3 or 4 feet away.
4. Place the microphone on the table approximately 12 to 18 inches between the user and the TV.
5. Connect the microphone cable to the mixer, and the mixer into the AVA-300's audio input.

NOTE

Consult the mixer manufacturer's instructions for full details on inputs and controls.

6. If you are using an EQ-unit, connect this on the audio path between the mixer and the AVA-300.
7. Place the video camera between the TV and the microphone, facing the user.

NOTE

The view of you that will be received at the remote end comes from this point, but your eyes will be looking at the center of the TV screen. Therefore, to give the best approximation of eye contact with the other party, you should place the camera as close as possible to your line of sight. To achieve this, mount the camera on top of the TV, or, if it's small enough, place it slightly in front of the TV. The less obtrusive the camera is, the closer it can be placed to your line of sight without obscuring important regions of the TV picture. As an example, the Canon VC-C1 camera has small dimensions and remotely controllable pan, tilt, zoom, and focus. Figure G.1 illustrates proper equipment positioning.

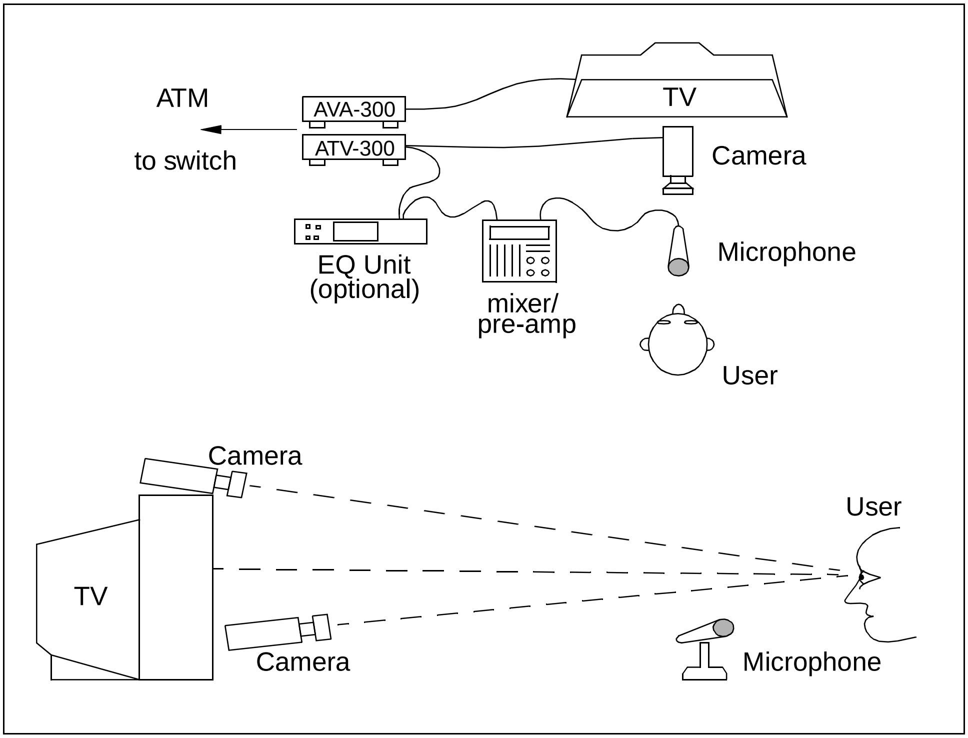

Figure G.1 - Audio/Video Equipment Placement

G.2.1 Setting the Volumes

Setting the proper volumes between two audio conferencing sites is an essential step to guaranteeing a successful and useful audio connection. The following procedure assumes two remote sites, called A and B. To set the proper audio volumes, do the following:

1. Turn the sound completely down on A's TV.
2. With user A speaking naturally into his or her microphone, increase the volume on A's pre-amp and B's TV until A can be clearly heard by B at a normal conversational volume.

NOTE

User B will have to gesture over the video channel to indicate whether the volume is too high or too low (i.e., "thumbs up" and/or "thumbs down").

3. If user B finds there is too much noise in the output, he or she should try increasing the pre-amp gain while user B decreases the volume on his or her TV.

NOTE

Noise is introduced into the signal when its amplitude is low, so amplifying the signal as early as possible increases the signal-to-noise ratio. If the signal is clipped and distorted (like the sound from an inexpensive stereo system being played too loud,) the pre-amp gain is too high and the AVA-300 is being overloaded. Lower the pre-amp gain to rectify the problem.

You should leave the AVA-300 audio gain at its default value of 8; this setting is adequate for most purposes.

4. Take note of the current settings corresponding to the path from site A to site B, turn the sound completely down on B's TV, and repeat steps 1 through 3 for the path from site B to site A.
5. Turn the sound back up on B's TV to its previous volume.

   At this point in principle, user A and user B can hear each other clearly. However, once both audio channels are enabled simultaneously, the system becomes more complex because of acoustic feedback, discussed in the next section.

G.2.2 Acoustic Feedback

In the simplest scenario, acoustic feedback occurs when sound leaving a loudspeaker enters the microphone that is driving the loudspeaker, causing the sound to circulate through the system many times. Figure G.2 illustrates simple feedback.

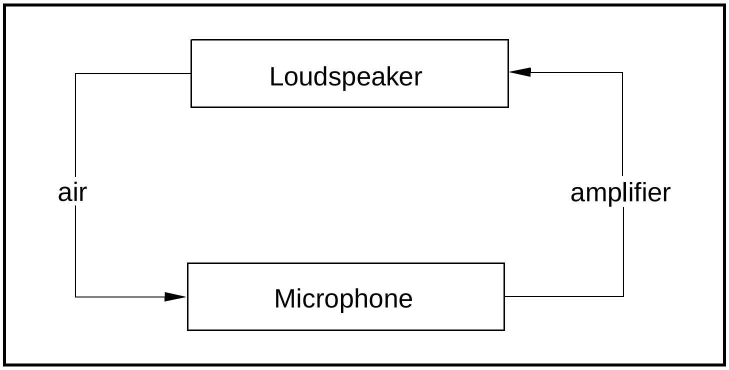

Figure G.2 - Simple Feedback

In a bidirectional, two-point teleconference, the setup is more complicated since the loudspeakers do not leak into microphones that drive them, but into the microphones driving the loudspeakers at the remote end. In this case, the circuit involves acoustic feedback at both air gaps. It follows that preventing acoustic feedback across either one of the air gaps will stop the system from experiencing feedback. Figure G.3 illustrates the feedback path in a two-point teleconference carried over ATM.

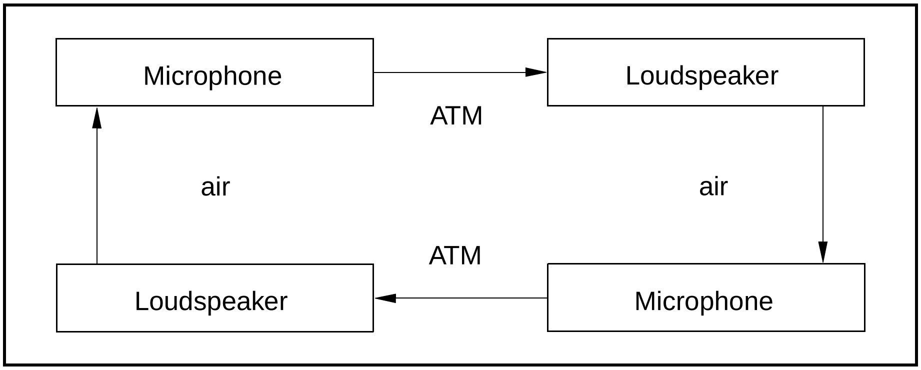

Figure G.3 - Feedback Path in a Two-Point Teleconference

Feedback presents itself to the listener in two forms, *howlround* and *reverberation*. These characteristics, as well as *echo,* are discussed in the following sections.

G.2.2.1 Howlround

Howlround occurs when the circuit gain for a frequency or range of frequencies is greater than one. Speaking into the microphone produces a noise of rising amplitude that eventually becomes just a single tone. Without intervention, it quickly reaches deafening amplitudes and may damage audio equipment.

G.2.2.2 Reverberation

Simply ensuring that the circuit gain be less than one is not enough to avoid feedback. If the circuit gain is close to, but less than, one, certain frequencies will reverberate. Reverberation is heard as a “muddy” echo that takes several seconds to dissipate once the subject has stopped speaking. Although this feedback goes away on its own, it makes communication difficult.

G.2.2.3 Echo

If the circuit gain is reduced sufficiently, each listener will be able to hear the sound of their own voice in their loudspeaker. In this situation, the cycle of feedback has been broken as the signal may only cross the air gap at the far end before it is silenced. In certain situations, such an echo may be useful (as in the public telephone system), since it gives each user the sensation that they are being heard at the other end. But if it is too loud or the delay is too long, it may become a distraction. Since two-point teleconferencing includes visual as well as audible cues to both users, echo is not necessary and is best eliminated.

G.2.3 Avoiding Feedback

To set up a successful two-point teleconference, the gain must be set low enough to avoid both howlround and reverberation, yet high enough to allow both users to hear each other at a reasonable volume. Since avoiding feedback requires a reduction of circuit gain, a limit is essentially placed on the sum of the volume controls at each side. This means that if user A decides to turn his or her volume up high to make listening easier, user B finds it difficult to use high volumes without feedback and must therefore strain to hear at a low volume. Collaboration between remote sites to achieve ideal volume settings is an important part of establishing a successful conference.

There are additional ways to reduce feedback. An equalizer, or EQ unit, can de-emphasize those frequencies that are not present in speech, preventing feedback at those frequencies. Typically, speech occupies the 250Hz - 8KHz band of the audio spectrum, but best results are obtained by progressively attenuating all frequencies outside the 800Hz - 1.6Hz range, with maximum attenuation (-16dB) applied outside the 250Hz - 8KHz band, as shown in Figure G.4.

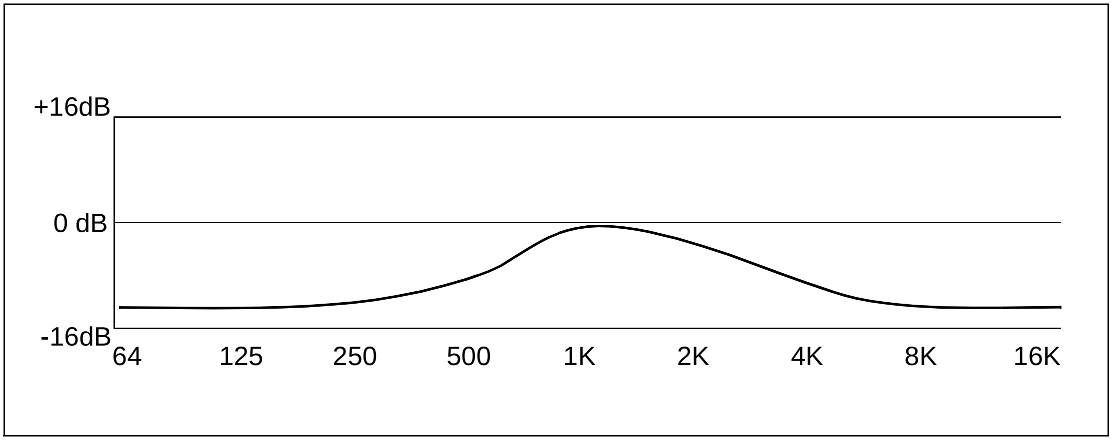

Figure G.4 - Typical EQ Curve for Feedback Reduction

Many EQ units, such as the Behringer Ultra-Curve, provide feedback eliminators. These signal processors constantly analyze the incoming signal for signs of feedback and quickly apply a narrow-band attenuating filter over the suspect frequencies whenever feedback is detected. See Section G.3 for details on configuring the Ultra-Curve.

G.2.4 Choosing Microphones

Choosing a quality microphone can greatly aid in reducing feedback. The two main criteria for choosing a microphone are the frequency response and the directional response characteristic.

Budget microphones tend not to be very consistent in their sensitivity and may reproduce sounds at some frequencies at much greater levels than others. At such frequencies, the microphone can be prone to feedback even though the gain over the frequencies is not be very high.

More expensive microphones tend to have a much flatter response, allowing the gain to be turned up across the whole audio spectrum without any peaks of extremely high gain to initiate feedback.

Professional microphones designed for relatively low levels of sound are much more sensitive and have much greater clarity than less expensive, robust microphones designed to tolerate sound levels ranging from quiet conversation to loud, live music.

Many microphones are of the *cardioid* (heart-shaped) variety, so named because their directional response characteristic looks like an inverted heart. For optimum results, the loudspeaker should be placed directly behind the microphone where its sensitivity is least; this corresponds to the dip in the `heart'. This can be difficult to achieve, since the loudspeaker placement may not fall within the null zone.

Other microphones have highly directional characteristics, making their response to sounds originating from behind the plane of the diaphragm relatively smaller. These *hypercardioid* microphones are well suited to a single stationary user.

Figure G.5 shows how the sensitivity varies with the angle between the source and the front of cardioid and hypercardioid microphones. The top of each graph represents the front tip of the microphone and the bottom represents the handle.

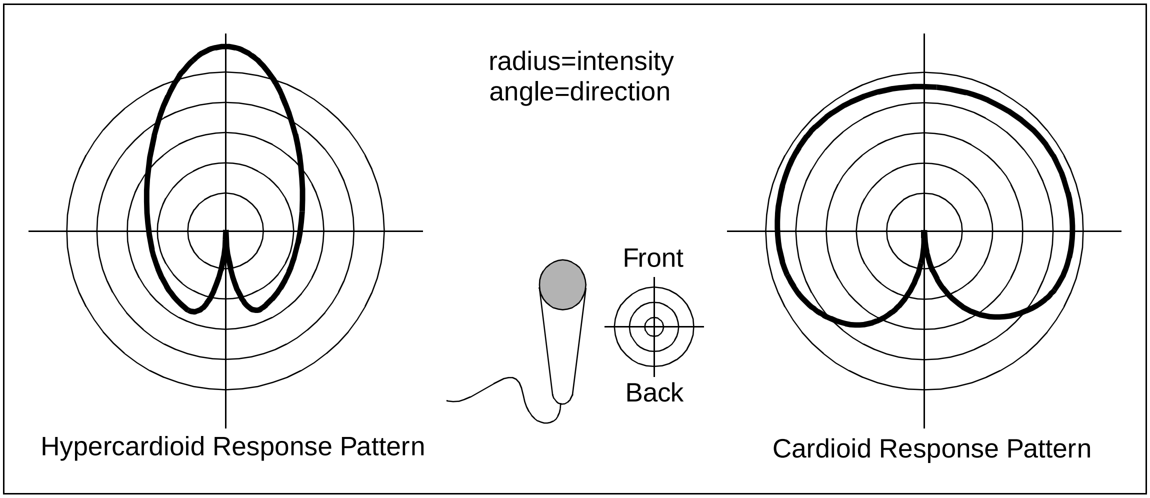

Figure G.5 - Microphone Sensitivity Response Graphs

G.2.4.1 Recommended Microphones

What follows is a selection of recommended microphones whose characteristics were compared against an easily-obtained, general-purpose budget model. All are of the back-electret condenser capsule design.

G.2.4.1.1 SONY ECM-Z3 Directional Microphone

This microphone has response characteristics that are similar to the budget microphone and in particular it has a low signal sensitivity.

G.2.4.1.2 Sennheiser ME64 Cardioid Capsule (requires K6 powering module)

This microphone produces sound of exceptional clarity. The signal strength is much greater and much flatter across the spectrum than the budget microphone. The sound is noticeably more directional, with the system is much less prone both to howlround and reverberation. Sound sources behind the plane of the capsule (e.g., the loudspeaker) were almost inaudible, but users in front of the plane were clearly audible even at very oblique angles. Multiple users could comfortably share a single microphone if seated around it properly.

G.2.4.1.3 Sennheiser ME66 Short Gun Capsule (requires K6 powering module)

This microphone virtually shares all of the characteristics of the ME64. The directionality is slightly greater, making this model better suited to a single user or users seated further away.

G.2.4.1.4 Crown PCC160 Pressure Zone Hypercardioid Microphone

Although the PCC160 is technically a hypercardioid model, the placement of the capsule gives it a more hemispherical (omnidirectional) response. When tested in the same environment as the other microphones, it received a substantially higher signal from the loudspeakers and was either too quiet or generating noisy feedback. An ideal placement is between the users on a conference table with the loudspeakers behind them.

G.2.5 Choosing Loudspeakers

Choosing loudpeakers is also a very important part of establishing a successful audio conference. used. After testing a number of TVs and a domestic stereo amplifier with a pair of hi-fi loudspeakers, the best results were obtained from a high-quality large-screen Sony television. Its loudpeakers point to the side and slightly back, and although high volume settings must be used, the clarity was good and the noise level low. The integrated speakers are especially convenient, doing away with a more complex set up requiring a separate amp and speakers. Smaller television loudpeakers seem slightly inadequate, creating distortion at higher volumes. A TV equipped with NICAM Stereo (in Europe) or Hi-Fi Stereo (in the U.S.) is highly recommended.

G.2.6 Other Equipment Placement Issues

G.2.6.1 Multiple Users per Microphone

Strategically arranging two or three users around a single cardioid or hypercardioid microphone is easy with a little trial-and-error. Participants closest to the microphone generally come across louder, while those farther away or at oblique angles tend to be quieter.

G.2.6.2 Multiple Microphones per Endpoint

It is possible to mix several microphones at each end to provide the audio feed for the remote site. Consult your mixer manufacturer’s User’s Manual for proper microphone mixing, as well as the information in the following sections:

G.2.6.2.1 Stereo

You can create true stereo effect by arranging two microphones in a triangle with the apex at the user(s). Place the microphones two to four feet apart depending on the number of users. If you have multiple users at the remote site, it is easy to create the illusion of spatiality with each of the users’ voices seeming to come from a different direction. Figure G.6 illustrates how spatiality relates to speaker positioning.

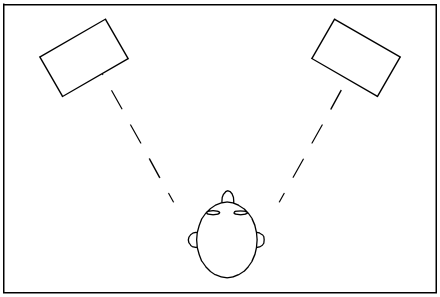

Figure G.6 - Positioning Loudspeakers for Illusion of Spatiality

Audio Conference Setup

G.2.6.2.2 Large Conferences

Two or more microphones arranged around a table with groups of users sharing a microphone is an easily-configured solution for large conferences. The only limits include the number of microphone inputs to the mixer and the maximum number of people that can comfortably share each one.

G.2.7 Other Audio Conferencing Tips

G.2.7.1 Walls

Be careful when placing users and microphones close to walls. Walls tend to reflect sound and can cause or accentuate feedback. Experiment with your speaker and microphone placements, and factor in the effect any walls that are close to your equipment may have on your conference's outcome.

G.2.7.2 Low Background Noise

An environment with low background noise is preferable to one with computers or other electronic hardware creating a constant hum. Such vibrations can be a particular nuisance if they are conducted through solids to the microphone; for this reason, well-insulated microphone stands are highly recommended if you attempt to conference in such an environment.

G.3 Configuring the Behringer Ultra-Curve

This section should be used in conjunction with, not in place of, the Ultra-Curve's User's Manual. The Ultra-Curve has two modes, EQ and RTA. EQ (or Equalization) mode allows the frequency response of the device to be altered; it is recommended that you use this mode.

RTA (or Real-Time Analyzer) mode shows an animated amplitude spectrum of the source signal. A typical use of RTA mode is for soundchecking, and for this purpose the device outputs a high amplitude pink-noise signal in this mode. Since this signal can damage hi-fi equipment if enabled at high volumes, the first step is to disable RTA mode:

1. Press `set up` twice, use the cursor keys to highlight `RTA LOCK`, and then click softkey `A` until `RTA LOCK ON` is displayed.
2. Click `EQ` to enter `EQ` mode.
3. The next step is to set the frequency response curve. Start at one end of the screen and use the `left` and `right` keys to move between selected frequencies. At each frequency, use the `up` and `down` keys to set the desired amplification or attenuation, using the values provided by the graph in Figure G.4 as your guide.

NOTE You can make the keys cursor twice as fast by holding one key down while pressing the opposite key; e.g., holding `left` then pressing `right` will make the cursor move to the left at double speed.

The next step is to enable the feedback eliminators. These filters can operate in one of two modes. In the first, you specify a narrow band which you suspect is where feedback is likely to occur. In the second mode, the device constantly scans the entire spectrum for suspected feedback, and it will set the feedback bands automatically. In each case a sharp attenuation filter is applied over these bands.

4. Set the feedback eliminator to `automatic` mode. Press `EQ`. Next, press softkey `B (FB-EL)` to display the feedback eliminator screen. There are three separate eliminators. Press softkeys `B`, `C`, and `D` (all labelled `FB-EL`) until the letter `S` is visible against each filter. Press `A (OK)` to return to the `EQ` screen.

   Once you have set the parameters, it is a good idea to save them for future use.

5. Press `D` (`Edit`). Next, press `A (Program Administration)` and then softkey `C (ABC...)` to get the `name` screen. By using the cursors and the `B (Next character)` softkey, choose a name for the curve you have defined. Press `A (OK)` to proceed. Next, press `A (Program Administration)` again, then `B (Save)`. Use the `up` and `down` keys to choose a slot (between `1` and `100`) in which to save the curve, then press the `OK` softkey.

NOTE

If there is already a program stored in that slot, you are prompted to that effect. Press `A (OK)` to overwrite, or `B (Cancel)` to leave the contents of that slot unchanged.

6. Insert the inputs from the mixer and connect the output to the AVA-300.

   You have successfully configured the Ultra-Curve.

G.4 Audio/Video Equipment Part Numbers and URLs

ForeRunner AVA-300 and ATV-300 Multimedia Codecs

`http://www.fore.com/products/video/index.html`

Sony KV-2375U NICAM Stereo TV

A European 27” Trinitron TV with flexible inputs & outputs, NICAM audio, and S-video inputs; Note the that the U.S. model KV-23V75 has fewer features - try the KV-32S22 instead.

`http://www.sel.sony.com/SEL/consumer/ss5/tvdsswebtvrtm/trinitronrtmcolortv27to32inch/kv-27v35.html`

Canon VC-C1 MkII Communication camera

Produces a sharp video picture, and can zoom, focus, pan and tilt with speed and precision; all of the camera’s features can be remotely controlled via an IR or by a PC.

`http://www.usa.canon.com/visualcomm/vcc1.html`

Spirit (by Soundcraft) Folio Notepad

Studio-quality mixer with 4 microphone inputs (XLR or 1/4” jack) and 2 Stereo (RCA) inputs, independent 2-band EQ, 50db gain pre-amps, and phantom powering.

`http://www.spirit-by-soundcraft.co.uk/brochures/notepad.html`

Behringer Ultra-Curve DSP-8000 EQ/RTA

A 31-channel programmable EQ unit with a spectral analyzer, feedback eliminators, and other digital effects.

`http://www.behringer.de/`

SONY ECM-Z3 microphone

A budget-priced, miniature directional back-electret conference microphone.

URL not available; see `http://www.sel.sony.com/SEL/consumer/ss5/accessories/camcordermicrophones/ecm-k57.html`

Sennheiser ME64 cardioid capsule

A professional cardioid capsule; requires K6 or K6P powering module.

`http://www.sennheiser.com/bm_1d004.htm`

Sennheiser ME66 short gun hypercardioid capsule

A professional hypercardioid capsule; requires K6 or K6P powering module.

`http://www.sennheiser.com/bm_1d006.htm`

Sennheiser K6 powering module

Battery / phantom powering module for ME capsules.

`http://www.sennheiser.com/bm_1d001.htm`

Sennheiser K6P phantom powering module

Phantom powering module for ME capsules.

`http://www.sennheiser.com/bm_1d002.htm`

Crown PCC160 pressure zone hypercardioid microphone

A stage-floor boundary microphone.

`http://www.crownaudio.com/pcc.htm`

APPENDIX H Windows NT/95: Manual De-install

If you have followed the software de-installation procedures in Section 5.3 and are unsuccessful, a manual de-installation may be required.

CAUTION

It is extremely important that you be familiar with the Windows NT/95 platforms when performing the operations detailed in this appendix. If not, consult your systems administrator for assistance or you risk placing your machine in an unusable state.

To perform a manual de-installation, do the following:

Use `regedit.exe` to remove the following keys from the registry:

In `HKEY_CLASSES_ROOT:`

`.sva` (NT only)

`svactrlfile` (NT only)

In `HKEY_LOCAL_MACHINE:`

`Software/Microsoft/Windows/CurrentVersion/UnInstall/ForeThought SVA-5.0`

`Software/Microsoft/Windows/CurrentVersion/App Paths/svc-rtds.exe`

`Software/FORE Systems, Inc./ForeThought SVA-5.0` (and all subkeys)

In Windows NT only:

Use the `Environment` tab of the `System Properties` box to remove the `NRLTRADERS` variable and remove the SVA directory from the `PATH` variable.

In Windows 95 only:

Remove the `NRLTRADERS` line from `autoexec.bat` and the SVA `bin` directory from the `PATH` line.

In Windows Explorer, remove the following files & directories:

NOTE

The location of these files is subject to the values you specified at install time; the locations shown below are the defaults. If you installed the SVA software or Netscape in another location, you must go to that location.

```
c:/Program Files/FORE Systems, Inc.
c:/winnt/profiles/*/Start Menu/Programs/ForeThought SVA-5.0
c:/Program Files/Netscape/Communicator/Plugins/npnpatch.dll
c:/Program Files/Netscape/Navigator/Plugins/npnpatch.dll
c:/Program Files/Microsoft Internet/Plugins/npnpatch.dll
c:/Program Files/Plus!/Microsoft Internet/Plugins/npnpatch.dll
```

Glossary

AAL (ATM Adaptation Layer) - the AAL divides the user information into segments suitable for packaging into a series of ATM cells. There are several types of AALs in use.

ANSI (American National Standards Institute) - a private organization that coordinates setting and approving US standards. It also represents the United States in the International Standards Organization.

ATM (Asynchronous Transfer Mode) - a transfer mode in which the information is organized into cells. It is asynchronous in the sense that the recurrence of cells containing information from an individual user is not necessarily periodic.

ATM Forum - a non-profit international industry consortium whose purpose is to accelerate rapid convergence on interoperability specifications based on international standards and promote industry cooperation.

ATV-300 - a dedicated unit for receiving and decoding the digital streams generated by an AVA-300 or other compatible device. The unit is suitable for situations where high-quality audio and video output signals are required or where it would be unsuitable to place a desktop computer.

AVA-300 - digitizes audio and video signals for direct transmission onto an ATM network. The digital streams generated are capable of being received and processed by workstations and ATV-300s.

Brightness - the amount of white (as compared to the amount of red, green and blue) in a color. On a monitor this translates into the amount of pure light in a color. For light sources and signals that transmit color, the brightness component is also called luminance.

Cell - an ATM Layer protocol data unit (PDU).

Cell Header - ATM Layer protocol control information.

Chrominance - the color component of a composite signal or S-Video signal. Chrominance also refers to the color component of any image, as opposed to its grayscale value or luminance.

Composite - a composite video signal is one that combines the chrominance, luminance and sync signals on a single wire. The device that receives the composite signal must decode the various kinds of information in order to display an image.

Contrast - the ratio between the maximum and minimum luminance (brightness) values of a display.

ECM - the External Configuration Module is an option for the AVA-300 and ATV-300 that allows you to configure a permanent start-up video and audio stream configuration. If power is to the unit is lost and then restored, it will start executing with the configuration stored in its ECM. The ECM contents may be changed over the ATM network.

Frame Buffer - memory used to store an array of graphic image data. Each element of the array corresponds to one or more pixels in a video display.

JPEG - the Joint Photographic Experts Group (JPEG) gives its name to an ISO method for still image compression. The amount of image compression is a function of a chosen quality factor, the amount of high-frequency detail contained in an image, and the viewer's tolerance to the resulting visual loss.

Luminance - a measure of brightness, the monochrome component of a composite or S-Video signal.

MTU (Maximum Transmission Unit) - the largest unit of data that can be sent over a type of physical medium.

NTSC (National Television Systems Committee) - a color-encoding and decoding system for the transmission of video signals, which when square-pixel sampled is 640 pixels wide by 480 pixels high at 60 fields/second or Hz. This system is used in the United States and Japan.

PAL (Phase Alternate Lines) - a color-encoding and decoding system for the transmission of video signals, which when square-pixel sampled is 768 pixels wide by 576 pixels high at 50 fields/second or Hz. This system is used in most European countries.

Pixel - a picture element that is the basic unit of a graphic display. A location on the monitor screen that can be selectively turned on or off.

PVC (Permanent Virtual Circuit (or Channel)) - a circuit or channel through an ATM network provisioned by a carrier between two endpoints; used for dedicated long-term information transport between locations.

Rate-Matching - dynamically adjusting the frequency of a sampled media stream (usually by interpolation or decimation), to correct for slight differences between original sampling rate and playout device rate due to different, unsynchonized clocks.

SECAM (Sequential Color and Memory) - a color-encoding and decoding system for the transmission of video signals. This system is used in France and the former Soviet Republics.

SVA - the control software environment for the AVA-300 and ATV-300 that runs on a range of popular workstations. It is composed of two logical components, `svc-rtds`, and `svc-mgr`, and allows you to manage the display characteristics and interconnections of any number of devices at one time.

svc-rtds - SVC Real-time Display System. Part of the SVA software distribution that allows you to edit and display live, TV-quality, AVA-300-generated video streams on a stock workstation.

S-Video - "Separate" Video is a physical video interface that carries the luminance and chrominance information separately. The S-Video signal is the same as the composite video signal, except that the Y signal (luminance) is on a separate wire from the U-V signal.

UNI (User-Network Interface) - the physical and electrical demarcation point between the user and the public network service provider.

VCI (Virtual Channel Identifier) - the address or label of a virtual channel (VC).

Index

T

U

V

W